



CONFIDENTIAL

Installation and Configuration Guide for SAP RDP Integration Accelerator

Contents

PURPOSE OF THIS DOCUMENT	5
HIGH LEVEL SCOPE AND APPROACH.....	5
TECHNICAL PREREQUISITE FOR INSTALLING THE PACAKGE	5
SECURITY	5
COMPONENT FOR CUSTOMER INCIDENTS	6
USER ASSISTANCE - SAP RESPONSIBLE DESIGN AND PRODUCTION.....	6
What's New in RDP Integration Accelerator V6	6
INSTALLATION OF TRANSPORT REQUESTS ON SYSTEM	8
STMS to import transport files	8
Copy provided files.....	8
Upload Transport Files	8
Install Transport Requests.....	10
CONTENT AND PACKAGE HIEARCHY	12
STEPS AND PREREQUISITES FOR CONFIGURATIONS OF INTEGRATION ACCELERATOR	14
Prerequisites	14
RFC Connections for RDP replication reports	14
RFC Connections for RDP Packaging Data reports.....	18
RFC Connections for RDP Packaging fee reports	19
Configuration settings for services.....	19
Maintain Email Notification Recipients - optional.....	22
Add Source ID	22
Create and activate Change Pointers	23
CONTENT OF RDP INTEGRATION PACKAGE	25
HOW TO REPLICATE DATA TO RDP.....	26
RDP ERP Integration reports.....	27
General Approach for Loading Data.....	27
Reports with Specific Load Functionality	28
Loading Configuration data	29
Loading Incoterms.....	30
Loading Organization data	31
Loading Batch data	31
Loading Supplier data	32
Loading Customer and Customer Extension data.....	34
Loading Product and Product extension data	35
Loading Deliveries.....	37
Loading Material documents.....	38
Loading Billing Document Items and Supplier Invoice Items.....	42
Loading Inventory.....	45
Extensibility for RDP ERP Integration reports	47
Sample Implementation.....	50

Badi Implementation Guidelines	51
Product and Customer extensions.....	53
Email notification adaptation.....	58
Excel documents processing adaptation	58
Resending failed payloads	59
RDP Packaging Data Integration reports	60
Creating recommended classes and characteristics	61
Upload Packaging Elements data from PLM.....	62
Class overview	62
Extensibility	63
Last change information	64
Upload Packaging Compositions from PLM	64
Class overview	64
Access the Specifications data	65
Upload Packaging Compositions from Recipe Management.....	65
Class overview	65
Access the Recipe Management data	66
Uploading Packaging Compositions from Bills of Material.....	66
Report parameters:	66
Class Overview	68
Uploading Packaging elements from Material Classifications.....	70
Report parameters	70
Class Overview	71
Uploading Packaging Composition Items from Handling Units	72
Report Parameters	74
Class Overview	75
Extensibility in packaging data reports	75
VPCOE/BADI_UPH_CUSTOM – technical documentation.....	76
Common class overview.....	79
Messages.....	79
API Version Selection for Packaging Data Integration	79
Packaging fee replication	80
Packaging fee retrieval	81
Packaging fee posting.....	82
Replication of Customer Exemptions	83
Replication of CompanyCode Exemptions	83
Replication of Customer Roles.....	84
Extensibility in Packaging fee reports.....	84
/VPCOE/BADI_PCKF_CUSTOM – technical documentation.....	84
HOW TO CHECK REPLICATION RESULTS	86

Results on the screen	86
Application log results	86
KNOWN TECHNICAL ISSUES AND TROUBLESHOOTING.....	87



PURPOSE OF THIS DOCUMENT

This document provides details of the installation and configuration steps for the Integration accelerator consulting package for RDP integration into ECC EHP7 or S/4 HANA 1909/2021. This document is not a replacement for standard RDP product scope. For detailed product information, please refer to the standard help documentation of [SAP Responsible Design and Production](#).

If you are interested in the details of scope of the Integration Accelerator package, please refer to the Detailed Scope Document which comes with this package.

HIGH LEVEL SCOPE AND APPROACH

The SAP Responsible Design and Production Integration Accelerator aims to speed-up the implementation of SAP RDP with ECC 6.0 EhP7 or with S/4 HANA 1909/2021.

The offering provides an ABAP based integration package, which comes with prebuilt integration content, which can be deployed to act as a fast-starting point for customers to integrate their RDP solution to their backend SAP system. The integration content helps in selecting, extracting and pushing the relevant data from the backend tables into the cloud based RDP solution via the [RDP push API interfaces](#).

This document describes the steps for installing and configuring the RDP Integration Accelerator Package.

TECHNICAL PREREQUISITE FOR INSTALLING THE PACKAGE

The below technical software component is the minimum version prerequisite for successfully installing and using the accelerator package:

SAP ERP	ECC 6.0 EHP 7
SAP_BASIS	740 Package Level 0019

Below is the minimum version for PLM software component required for using the PLM package:

SPLMWUI	747 SP Level 0016 Support Package: SAPK-74716INPLMWUI
---------	--

In case [Packaging Composition from HU report](#) is used from a logistics system, please consider that RDP Integration Accelerator can be deployed on customer logistics system if it is a NetWeaver systems with ABAP application server.

SECURITY

All replication endpoints are secured with TLS (https) to ensure encryption of the customer data being sent.

Authorization is using XSUAA and the OAuth 2.0 client credentials grant.

Using the RDP service broker on BTP, customers can create a service instance that allows to get an access token for replication.

SSL certificates/handshake to build the trust between the systems is a prerequisite for data replication.

COMPONENT FOR CUSTOMER INCIDENTS

For any technical assistance for RDP Integration Package, you can create support tickets accessing [SAP ONE Support Launchpad](#) using Ticket Component: SV-COE-VPT-RDP.

USER ASSISTANCE - SAP RESPONSIBLE DESIGN AND PRODUCTION

For the user assistance for SAP Responsible Design and Production, see [SAP Responsible Design and Production - SAP Help Portal](#).

SAP Responsible Design and Production provides public API for data replication. To be able to push data from your system you need to configure the replication in SAP Responsible Design and Production. Please refer to this [link](#) to get more information on the steps to be performed.

To access the RDP API documentation, go to [API Reference | Replication | SAP API Business Hub \(ondemand.com\)](#).

For more information on the API elements, see [API for data Replication](#) and [API References](#)

SAP API Business HUB RDP Overview page: [Overview | Replication | SAP API Business Hub](#)

Note:

To get started with SAP Responsible Design and Production use the standard procedures for the SAP BTP, Cloud Foundry environment. For an overview of the required steps, see [Getting Started in the Cloud Foundry Environment](#).

The following prerequisites must be fulfilled

- An SAP BTP global account
- Members are added to the SAP BTP account
- An SAP BTP Cloud Foundry subaccount is created and enabled

Subscribing to Services

Subscribe to the following services in the SAP BTP cockpit.

- SAP Responsible Design and Production

For instruction, see [Subscribe to Multitenant Applications Using the Cockpit](#).

For more information on the steps to be performed on BTP global account for setting up the SAP

What's New in RDP Integration Accelerator V6

The **V6 release** of the SAP RDP Integration Accelerator introduces enhancements to improve data replication, performance, and additional attributes added or removed. This version includes new features and refinements across multiple integration areas:

1. Customer Data Enhancements

- Updated sample **BAdI implementation** and **Excel upload functionality** to filter tax numbers based on SAP-defined tax categories, eliminating unsupported entries.
- **New selection filter:** Added **Company Code** as a filtering criteria.

2. Supplier Data Enhancements

- **New selection filter:** Added **Company Code** as a filtering criteria.

3. Product Data Replication Enhancements

- Introduced additional attributes for **Quantity Dimensions**, including:
 - **Net Weight**
 - **Gross Weight**
 - **Weight Unit of Measure**

4. Batch Replication Enhancements

- Implemented additional logic to determine **packaging composition** in batch replication.

5. Delivery Data replication Enhancements

- Performance optimizations for faster and more efficient data replication.
- Support for **Delivery Batch Split** scenarios.
- Handling of **Catch Weight Material** scenarios.

7. Packaging Composition Enhancements

- Added support for ending validity of a product in a packaging composition - **Valid To Date assignments**. Sample implementation is offered in the sample BAdI implementation.
- Improved handling of material number **selection criteria restrictions** to enhance flexibility.
- **New exclusion rule:** Data with a **deletion flag** will now be excluded from replication.

8. Packaging Elements Enhancements

- **New exclusion rule:** Data with a **deletion flag** will now be excluded from replication.
- Introduction of **Reusability** functionality (isReusable API attribute)

9. Packaging Data V2 APIs

- Enhanced support for **Version 2 RDP APIs** for both Packaging elements and Packaging compositions.
- Customers can choose if they want to use V2 or V1 of APIs, by adjusting the services paths
- Sample BAdI implementations are available for both V1 and V2 of the APIs

10. Packaging Fee Replication Enhancements

- **New Sample BAdI Implementation for Packaging Fee Replication for CONAI.**
 - Support for retrieving **Customer Exemptions** and **Company Code Exemptions**.

- **Customer-Dependent Pricing Condition Replication:**

- **New parameter: Customer Role Dependent Fees** included in the packaging fee retrieval report.
- Retrieval of **Customer Role Replication**.

11. New functionality: RDP Classes & Characteristic Generation

- A new report to create recommended **RDP classes and characteristics**, supporting:
 - Packaging Elements in Material Classification
 - Packaging Compositions in BOM, classifying Pckg. Elmnts. into Main/Subordinate
 - Packaging Compositions and Main/Subordinate Packaging Elements in PLM SpecDB
 - Packaging Fee Data Classification
 - Customer Extension Data Classification

INSTALLATION OF TRANSPORT REQUESTS ON SYSTEM

Prerequisites:

Authorizations for the following transaction codes:

- CG3Z in client 000 or any other needed client to upload transport files (or authorization to execute Function Module ARCHIVFILE_CLIENT_TO_SERVER)
- STMS

STMS to import transport files

As a first step, the provided packages must be imported into your ECC or S/4 HANA system. Follow the below steps to import the necessary RDP Integration Accelerator packages.

Copy provided files

For installing RDP development package on the backend ECC or S/4 HANA system it is necessary to copy the provided files on the system on which SAP is installed. (Ex. **K900446.BP7** & **R900446.BP7**). Copy all the files we have provided to you, for General objects, RDP and/or PLM objects.

Please note the following sequence to install the transport requests (if otherwise stated in ReadMe.txt file when provided):

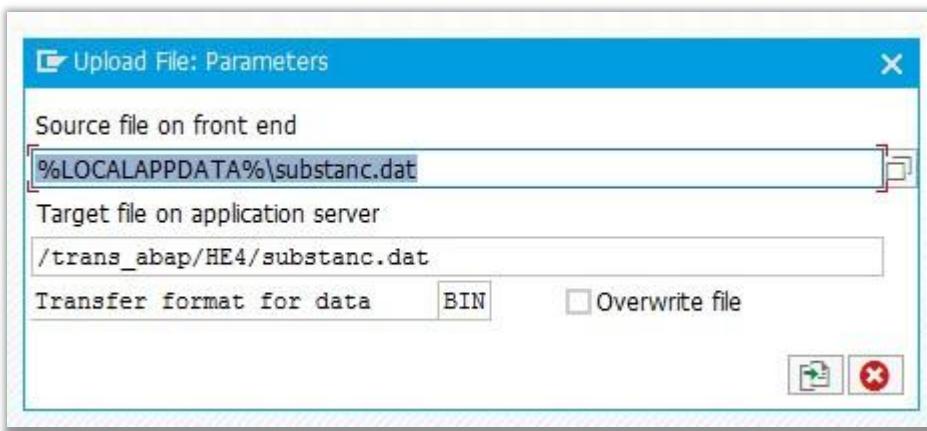
1. TR for General Objects
2. TR for RDP Objects
3. TR for PLM Objects

Please read the file ReadMe.txt shared together with the TRs files, explaining the TRs shared and sequence to be installed in more particular way.

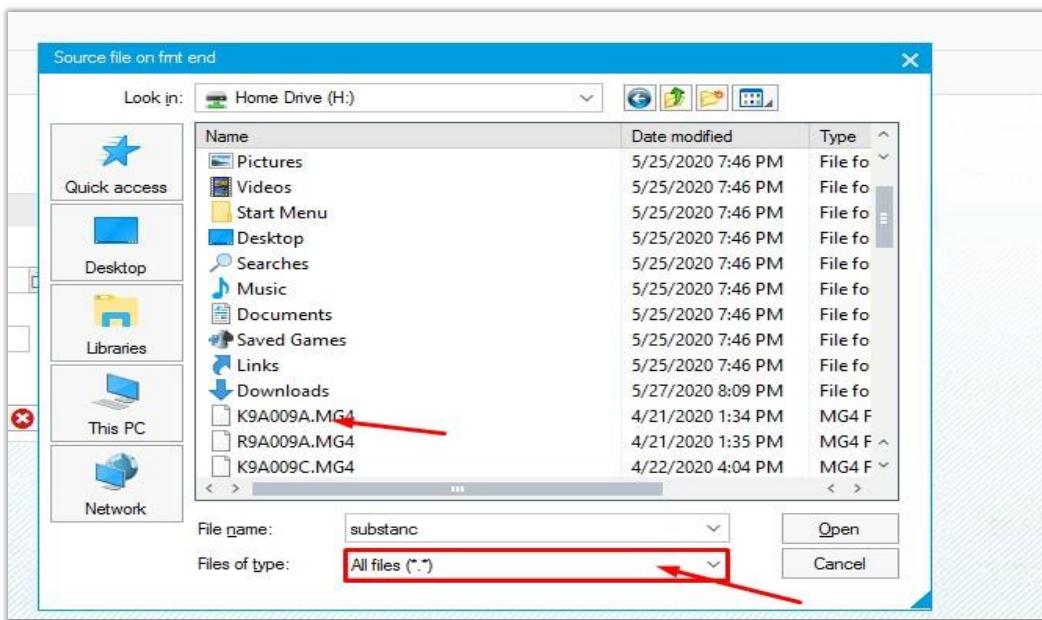
Note: the screenshots for uploading and installing the TRs are using sample TRs and are for your reference only.

Upload Transport Files

In your system, open necessary client and upload transport files on your system server. Open transaction 'CG3Z' (If transaction doesn't exist on the system see next step).



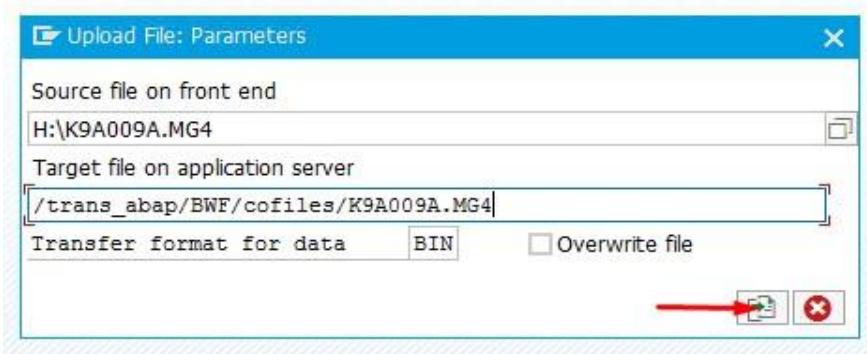
Choose transport files provided to you:



Enter details on source file and target file on your application server:

1. ./trans_abap/{system}/cofiles/{name transport file} (for file having first later 'K')
2. ./trans_abap/{system}/data/{name transport file} (for file having first later 'R')

Note: For details on path configured on your application server, use transaction 'AL11'.



Press upload. (Ex. upload cofiles on BWF system).

Test Function Module: Initial Screen

Debugging Test data directory

Test for function group OPTA
Function module ARCHIVFILE_CLIENT_TO_SERVER
Uppercase/Lowercase

Import parameters	Value
PATH	H:\K9A00AC.MG4
TARGETPATH	/trans_abap/MG4/cofiles/K9A00AC.MG4

If 'cg3z' transaction doesn't exist on your system, please follow below step:

- Use Function Group: **OPTA**, function module: **ARCHIVFILE_CLIENT_TO_SERVER** (**execute FM using transaction code SE37**)
Add the path where source files will be saved, and also the server path (target file on application server). Use Uppercase/Lowercase flag.
- Enter details on source file and target file on application server:
 1. /trans_abap/{system}/cofiles/{name transport file} (for the file having first later 'K')
 2. /trans_abap/{system}/data/{name transport file} (for the file having first later 'R')
- Execute above steps for all provided files, for general objects and for RDP and PLM objects.

Note: For details on path configured on your application server, use transaction 'AL11'.

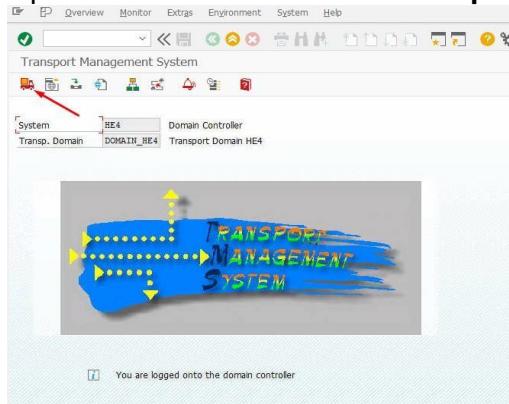
Install Transport Requests

Please follow below steps to install all transport requests provided.

Note: Please note the order to install the transport request (check also the ReadMe.txt file shared to you together with TRs, for any particular cases):

1. TR for General Objects (mandatory pre-requisite for all other packages)
2. TR for RDP Objects
3. TR for PLM Objects (only needed if you are integrating SAP PLM to RDP)

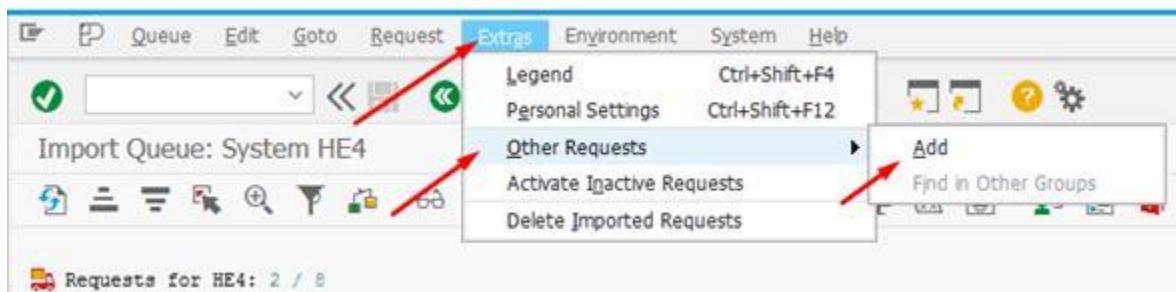
1. Open transaction 'STMS'. Click 'Import Overview (F5)'



2. Double-click on the entry with the name of the system.

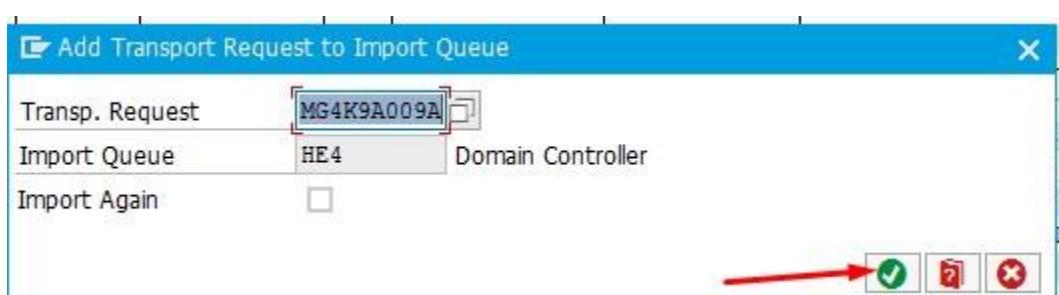
Queue	Description
CLS	Dummy System
HE4	Domain Controller
Σ	

3. Add new transport request on the queue.



4. Choose necessary transport request (since 'K9A009A.MG4' file has been uploaded, 'MG4K9A009A' entry should be selected).

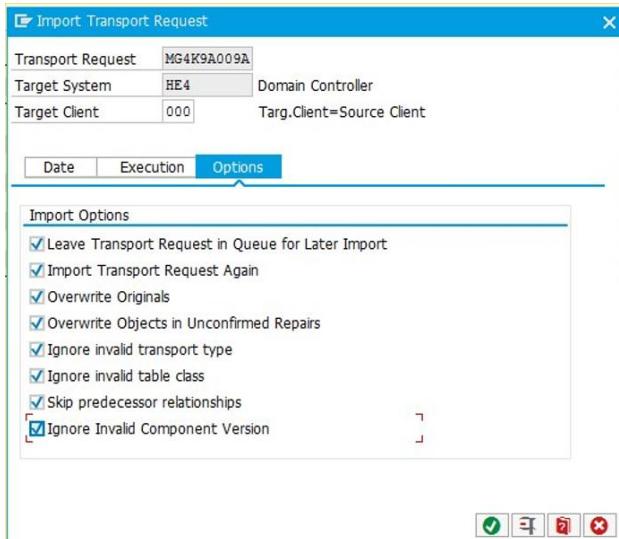
Request/Task
ECZK9A0006
ECZK9A000F
ECZK9A000G
ECZK9A000H
MG4K9A009A
MG4K9A009C
HF4K9A007P



5. Choose the entry with the name of the transport request and click 'Import request':

Number	Request	RC	Owner	Project	Short Text
1	ECZK9A0006				
2	ECZK9A000F				
3	ECZK9A000G				
4	ECZK9A000H				
5	MG4K9A009A				
6	MG4K9A009C				
7	HE4K9A008F				
8	HE4K9A008X				

6. Enter target client and select all check boxes and press Continue. On the next screen click ‘Yes’:



CONTENT AND PACKAGE HIERARCHY

The package provides the relevant code to select the required data from SAP backend system and replicating it to the RDP tenant, using standard HTTP endpoints.

The package includes the following integration components:

ERP Integration Reports

- **Organizational Data** – Replication of company structures, plants, sales organizations, distribution channels, and divisions.
- **Configuration Data** – Synchronization of unit of measure definitions, currency details, product hierarchies, movement types, and other configuration elements.
- **Master Data Replication**
 - Supplier
 - Customer
 - Product
 - Batches

Transactional Data Replication

- **Delivery Documents** including the **Billing Document Items**
- **Material Movements** including the **Billing Document Items & Supplier Invoice Items**
- **Inventory**
- **Incoterms**

Packaging Data Integration

- **Packaging Composition Replication**
 - Extracted from PLM, Recipe Development, and Bill of Materials (BOM).
- **Packaging Elements Replication**
 - Retrieved from PLM and Material Classification.

- **Partial Packaging Composition Item Replication**
 - Handling Units replication.
- **Creation of RDP recommended classes and characteristics**
 - create example classes and characteristics for storing packaging data in your SAP ECC / S/4HANA system based on Material Classification or Specification Database.

Packaging Fee Replication

- **Synchronization of Packaging Fees** – Replication from RDP back to ECC/S4H for creating pricing conditions and compliance reporting.
- **Customer Role Replication** – Replication of customer-specific role attributes from RDP to ECC/S4H for creating customer dependent/independent pricing conditions and material classifications
- **Customer Exemptions and Company Code Exemptions** – Replication of customer-specific and company code-specific exemptions from RDP to ECC/S4H for creating customer dependent/independent pricing conditions and material classifications.

Analytical and Configuration Reports

- **Report for Generating Recommended RDP Classes and Characteristics**
 - Packaging Elements in Material Classification
 - Packaging Compositions in BOM, classifying Pckg. Elmnts. into Main/Subordinate
 - Packaging Compositions and Main/Subordinate Packaging Elements in PLM SpecDB
 - Packaging fee data classification
 - Customer extension data classification

Error Handling and Logging

- **Error Monitoring and Troubleshooting Reports** – Comprehensive logging and troubleshooting capabilities for monitoring replication status and resolving data inconsistencies.

Package hierarchy and structures:

/VPCOE/CLOUD_INTEGRATION is the base package containing following sub-packages:

/VPCOE/COMMON_INTEGRATION containing common Integration objects

/VPCOE/RDP_INTEGRATION containing *RDP ERP* Integration objects/reports

/VPCOE/PLM_INTEGRATION containing RDP Packaging Data Integration objects/reports (only for PLM component)

/VPCOE/OBJ_INTEGRATION containing RDP Packaging Data Integration objects/reports from BOM and Material classification. It also includes the Handling Units replication report.

/VPCOE/RDP_PCKGFEET_INTEGRATION containing packaging fee Integration objects/reports and also customer role/ customer exemptions and company code exemptions replications from RDP back to ECC/S4H

STEPS AND PREREQUISITES FOR CONFIGURATIONS OF INTEGRATION ACCELERATOR

Once you have followed the above steps to import the transport packages, the next step is to configure the technical connectivity between the backend system and RDP tenant.

Prerequisites

- Setting up the replication services in SAP BTP Cloud Cockpit
Please follow [these steps](#) to set up the replication services and subscribe to RDP service
- Getting credentials of the SAP RDP service from your tenant:
 - Client Id
 - Client Secret
 - token endpoint URL
- Establish the SSL connection to SAP BTP
 - follow this link to establish connection: [Establish the SSL Connection to SAP BTP - SAP Help Portal](#)
 - Follow the instruction in note for downloading and importing BTP certificates: [2853519 - How to download the SSL certificate of SAP BTP - SAP for Me](#)
- Authorizations for the following transactions:
 - SM59 to create an RFC destination

For more information on all the steps required to be performed on BTP Cockpit and on token endpoint URL and client credentials, see [Push Data Replication](#).

The integration package invokes the RDP APIs via HTTPS connection from the backend system. To support this, several RFC connections has to be created on the backend system for RDP reports. The below steps have to be done to create and configure the RFC connections.

Note: For RDP Integration package there is no need to configure OAuth Client 2.0

RFC Connections for RDP replication reports

The following RFC destinations of type 3 – ABAP Connection must be created in the backend system. (Transaction code ‘SM59’)

- VPCOE_RDP_GENERIC
- VPCOE_RDP_OAUTH
- VPCOE_RDP_PLM
- VPCOE_RDP_PLM_PKG_ELMNT
- VPCOE_RDP_PCKF

To create a new RFC destination, carry out the following steps:

1. Open transaction SM59
2. Choose *Create*.

3. Enter name `VPCOE_RDP_GENERIC` for the RFC destination.
4. Select connection type G (HTTP Connection to External Server).
5. On the **Technical Settings** tab, enter the following data:
 1. **Target Host:** server endpoint URL (ex. {tenant-subdomain}.eu20.ce.sustainability.cloud.sap)
 2. **Port:** 443

RFC Destination VPCOE_RDP_GENERIC

Connection Test

RFC Destination	VPCOE_RDP_GENERIC
Connection Type	G HTTP Connection to External Serv
Description	
Description 1	RDP Integration: Generic Destination
Description 2	Path Prefix changes from VPCOE Report
Description 3	

Administration **Technical Settings** Logon & Security Special Options

Target System Settings

Target Host	0-integ-ecc.dev-eu20.ce.sustainability.cloud.sap	Service No.	443
Path Prefix	/api/replication/v1/Deliveries		

HTTP Proxy Options

Global Configuration	
Proxy Host	
Proxy Service	
Proxy User	
Proxy PW Status	is initial

6. On the **Logon & Security** tab
 1. Set **Do not use a user**.

Set **SSL to Active**.

RFC Destination VPCOE_RDP_GENERIC

Connection Test

RFC Destination	VPCOE_RDP_GENERIC	
Connection Type	G HTTP Connection to External Serv	Description
Description		
Description 1	RDP Integration: Generic Destination	
Description 2	Path Prefix changes from VPCOE Report	
Description 3		

Administration Technical Settings **Logon & Security** Special Options

Logon Procedure

Logon with User

Do Not Use a User
 Basic Authentication

User
 PW Status is initial

Logon with Ticket

Do Not Send Logon Ticket
 Send Logon Ticket Without Ref. to a Target System
 Send Assertion Ticket for Dedicated Target System

System ID Client

Security Options

Status of Secure Protocol

SSL Inactive Active
 SSL Certificate DEFAULT SSL Client (Standard) Cert. List

Authorization for Destination

Create another RFC destination for OAuth token following below steps:

1. Open transaction SM59
2. Choose **Create**.
3. Enter name VPCOE_RDP_OAUTH for the RFC destination.
4. Select connection type G (HTTP Connection to External Server).
5. On the **Technical Settings** tab, enter the following data:
 1. **Target Host:** authentication endpoint URL (ex: {tenant-subdomain}.authentication.eu20.hana.ondemand.com)
 2. **Path prefix:** /oauth/token
 3. **Port:** 443

RFC Destination VPCOE_RDP_OAUTH

Connection Test

RFC Destination	VPCOE_RDP_OAUTH
Connection Type	G HTTP Connection to External Serv
Description	
Description 1	RDP Integration: oAuth 2.0 destination
Description 2	
Description 3	

Administration Technical Settings Logon & Security Special Options

Target System Settings

Target Host	[REDACTED] authentication.eu20.hana.ondemand.com	Service No.	443
Path Prefix	/oauth/token		

HTTP Proxy Options

Global Configuration	
Proxy Host	
Proxy Service	
Proxy User	
Proxy PW Status	is initial

6. On the Logon & Security tab

1. Set **Basic Authentication**
2. Enter **username** and **password**
3. Set **SSL to Active** and choose SSL certificate to be used.

RFC Destination VPCOE_RDP_OAUTH

Connection Test

RFC Destination	VPCOE_RDP_OAUTH
Connection Type	G HTTP Connection to External Serv
Description	
Description 1	RDP Integration: oAuth 2.0 destination
Description 2	
Description 3	

Administration Technical Settings Logon & Security Special Options

Logon Procedure

Logon with User

- Do Not Use a User
- Basic Authentication

User	[REDACTED]
PW Status	saved

Logon with Ticket

- Do Not Send Logon Ticket
- Send Logon Ticket Without Ref. to a Target System
- Send Assertion Ticket for Dedicated Target System

System ID	[REDACTED]	Client	[REDACTED]
-----------	------------	--------	------------

Security Options

Status of Secure Protocol

SSL	<input type="radio"/> Inactive	<input checked="" type="radio"/> Active
SSL Certificate	DFAULT SSL Client (Standard)	Cert. List

Authorization for Destination

7. Save

Note: The token URL, username, and password required for authentication can be retrieved from the SAP Business Transformation Platform (BTP) Cockpit, as detailed in [the official documentation](#).

When testing the RFC connection **VPCOE_RDP_OAUTH**, you may encounter the error message:
{"error": "invalid_request", "error_description": "Missing grant type"}

This message can be disregarded, as the RDP Integration package handles the grant type

RFC Connections for RDP Packaging Data reports

Create a new RFC destination, carry out the following steps:

1. Open transaction SM59
2. Choose *Create*.
3. Enter name **VPCOE_RDP_PLM** for the RFC destination.
4. Select connection type G (HTTP Connection to External Server).
5. On the **Technical Settings** tab, enter the following data:
 1. **Target Host:** token endpoint URL (e.g. {tenant-subdomain}.eu20.ce.sustainability.cloud.sap)
 2. **Path Prefix:** /api/replication/v1/PackagingCompositions
 3. **Port:** 443
6. On the **Logon & Security** tab
 1. Set **Do not use a user**.
 2. Set **SSL to Active** and choose SSL certificate to be used.
7. Save

Create another RFC destination following the same steps above:

1. Enter name **VPCOE_RDP_PLM_PKG_ELMNT** for the RFC destination
2. On the **Technical Settings** tab, enter the following data:
 1. **Target Host:** token endpoint URL (e.g. {tenant-subdomain}.eu20.ce.sustainability.cloud.sap)
 2. **Path Prefix:** /api/replication/v1/PackagingElements
 3. **Port:** 443
3. Save

Note: the name of RFC destinations created for Packaging Data reports can differ, as the RFC destination is a selection parameter that has to be specified when executing the reports.

For more information on creating the RFC connections for Packaging Data Integration, please refer to this [link](#).

RFC Connections for RDP Packaging fee reports

Create a new RFC destination, carry out the following steps:

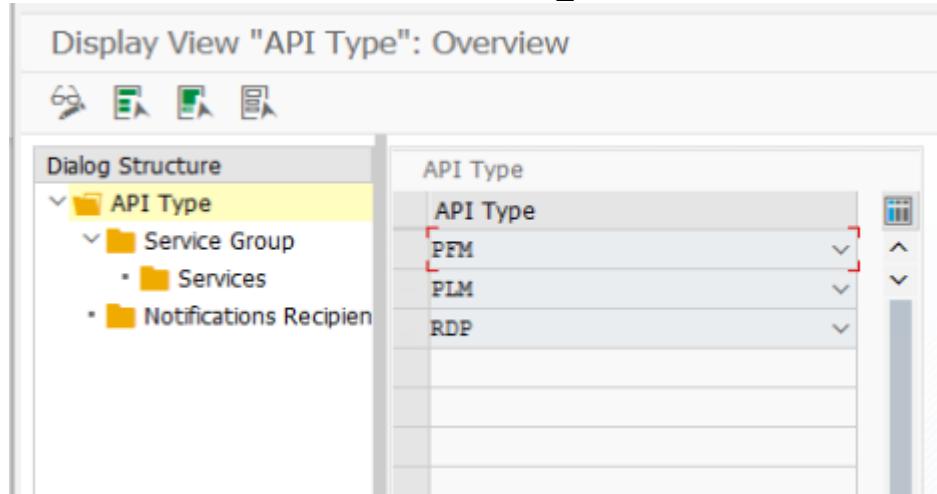
8. Open transaction SM59
9. Choose **Create**.
10. Enter the name `VPCOE_RDP_PCKF` for the RFC destination.
11. Select connection type G (HTTP Connection to External Server).
12. On the **Technical Settings** tab, enter the following data:
 1. **Target Host:** token endpoint URL (e.g. {tenant-subdomain}.eu20.ce.sustainability.cloud.sap)
 2. **Path Prefix:** /api/packaging-report/v1/
13. On the **Logon & Security** tab
 1. Set **Do not use a user**.
 2. Set **SSL to Active** and choose SSL certificate to be used.
14. Save

Configuration settings for services

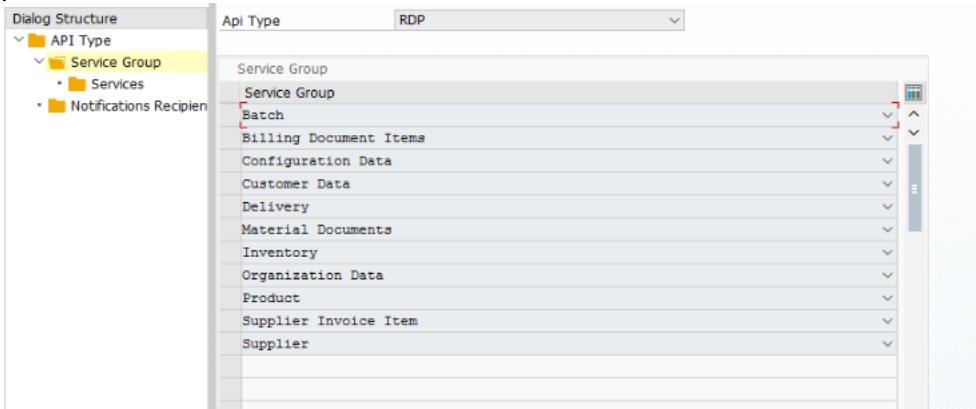
Maintain Service Path prefix and package size

In the following steps you will add the required configuration settings to the service groups/ service IDs .

Open transaction 'SM34'. Enter /VPCOE/RDP_SRV view cluster and click on 'Maintain' button.



Select API type 'RDP' and double-click on 'Service Group'. You will get a list with all RDP service groups:



Maintain the path prefix and package size.

Click on necessary Service Group and choose Services to maintain the path prefix or package size for it.

The maximum recommended package size to be used for all services is 1000 (except Material Documents where maximum recommended package size is 2000). Click on Save button to save all configurations.

Display View "Services": Overview

Service Id	Path Prefix	PkgSize	WPCount
Country	/api/replication/v1/Countries	500	0
Currency	/api/replication/v1/Currencies	500	0
Delivery Document Item Category	/api/replication/v1/DeliveryDocumentItemCategories	500	0
Delivery Document Type	/api/replication/v1/DeliveryDocumentTypes	1.000	0
Incoterms	/api/replication/v1/Incoterms	1.000	0
Movement Type	/api/replication/v1/MovementTypes	1.000	0
Product Group	/api/replication/v1/ProductGroups	1.000	0
Product Hierarchy	/api/replication/v1/ProductHierarchies	1.000	0
Product Type	/api/replication/v1/ProductTypes	1.000	0
Region	/api/replication/v1/Regions	1.000	0
UoM	/api/replication/v1/UnitOfMeasure	500	0
UoM Code	/api/replication/v1/UnitOfMeasureIsoCodes	500	0
UoM Dimension	/api/replication/v1/UnitOfMeasureDimensions	500	0

Display View "Services": Overview

Service Id	Path Prefix	PkgSize	WPCount
BOM	/api/replication/v1/PackagingCompositions	1.000	0
MCL	/api/replication/v1/PackagingElements	500	0
PLM	/api/replication/v1/PackagingCompositions	500	0
RCP	/api/replication/v1/PackagingCompositions	500	0
UPLOAD_DATA	/api/replication/v1/PackagingElements	500	0

WPCount represent the number of Work Processes to be used in parallel when data are sent to RDP for a specific service. WPCount is not editable for all the services except Deliveries and Material Documents.

Please note that transactional data reports (Material document, Delivery) are executing using multiple parallel work processes. The number of work processes to be used can be configured for those services. The maximum recommended number of work processes to be used in parallel is 5:

Display View "Services": Overview						
Dialog Structure		Service Group	DLVR	Api Type	RDP	
▼ API Type	▼ Service Group	Services				
▼ Services						

Display View "Services": Overview						
Dialog Structure		Service Group	DOC	Api Type	RDP	
▼ API Type	▼ Service Group	Services				
▼ Services						

Path prefix list for RDP API:

Batch	
Batch	/api/replication/v1/Batches/
Configuration Data	
Country	/api/replication/v1/Countries
Currency	/api/replication/v1/Currencies
Delivery Document Item Category	/api/replication/v1/DeliveryDocumentItemCategories
Delivery Document Type	/api/replication/v1/DeliveryDocumentTypes
Incoterms	/api/replication/v1/Incoterms
Movement Type	/api/replication/v1/MovementTypes
Product Group	/api/replication/v1/ProductGroups
Product Type	/api/replication/v1/ProductTypes
Product Hierarchy	/api/replication/v1/ProductHierarchies
Region	/api/replication/v1/Regions
UoM	/api/replication/v1/UnitOfMeasures
UoM Code	/api/replication/v1/UnitOfMeasureIsoCodes
UoM Dimension	/api/replication/v1/UnitOfMeasureDimensions
Customer Data	
Customers	/api/replication/v1/Customers
Customer Extensions	/api/replication/v1/CustomerExtensions
Delivery	
Delivery	/api/replication/v1/Deliveries
Material Documents	
Material Documents	/api/replication/v1/MaterialDocuments
Organization Data	
Company Code	/api/replication/v1/CompanyCodes
Plant	/api/replication/v1/Plants
Sales Organization	/api/replication/v1/SalesOrganizations
Distribution Channel	/api/replication/v1/DistributionChannels
Division	/api/replication/v1/Divisions
Batch	
Batch	/api/replication/v1/Batches
Product	
Product	/api/replication/v1/Products
Product Extensions	/api/replication/v1/ProductExtensions
Supplier	
Suppliers	/api/replication/v1/Suppliers
Supplier Extensions	/api/replication/v1/SupplierExtensions
Inventory	
Inventory	/api/replication/v1/Inventories

Incoterms	
Incoterms	/api/replication/v1/Incoterms
Billing Document Items	
Billing Document Items	/api/replication/v1/BillingDocumentItem
Supplier Invoice Items	
Supplier Invoice Items	/api/replication/v1/SupplierInvoiceItem

Path prefix list for Packaging Data APIs:

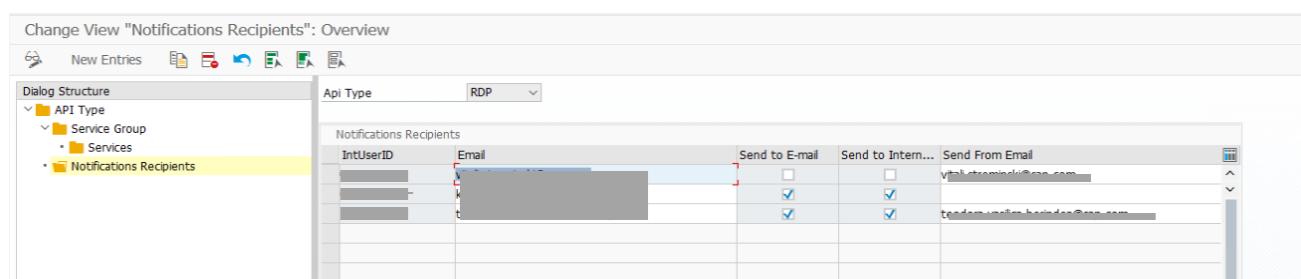
Packaging Data	
BOM	/api/replication/v1/PackagingCompositions
MCL	/api/replication/v1/PackagingElements
PLM	/api/replication/v1/PackagingCompositions
RCP	/api/replication/v1/PackagingCompositions
UPLOAD_DATA	/api/replication/v1/PackagingElements

In case user wants to use V2 of the packaging data API, the path prefix should be adjusted accordingly:

Packaging Data	
BOM	/api/replication/v2/PackagingCompositions
MCL	/api/replication/v2/PackagingElements
PLM	/api/replication/v2/PackagingCompositions
RCP	/api/replication/v2/PackagingCompositions
UPLOAD_DATA	/api/replication/v2/PackagingElements

Path for Packaging composition items (from handling units) is defined in the entity processor class /VPCOE/CL_UPH_TM_HU attribute MC_HU, value '/PackagingCompositionItems'. This value can be overwritten if it's necessary.

Maintain Email Notification Recipients - optional



In case of any failed replication in background mode, if email addresses are maintained, emails with the status of failed replications are sent to maintained users. There are two options for sending emails: to ECC internal box or directly to email addresses. Please note that in order emails to be sent out from ECC system, all basis configurations need to be done to allow this.

Add Source ID

Open transaction 'stvarv'. Create new entry:

Name: /VPCOE/RDP_SOURCE_ID

Display Table TVARVC: Selection Variables		
<input checked="" type="checkbox"/> Individual maintenance		
Contents of Table TVARVC		<input type="checkbox"/> Include changed entries in transport request
Parameter	Selection Options	
Name	Val.	Case...
/VPCOE/RDP_SOURCE_ID	ECC	<input type="checkbox"/>
SAP_FAGL_MCA_GENL_LOG_DAYS	10	<input type="checkbox"/>
SAP_SCMA_DETAIL_LIST		<input type="checkbox"/>
SAP_SCMA_FISC_YEAR		<input type="checkbox"/>
SAP_SCMA_PERIOD		<input type="checkbox"/>
SAP_SCMA_POST_DATE		<input type="checkbox"/>
SAP_SCMA_POST_PERIOD		<input type="checkbox"/>
SAP_SCMA_SELV_PROJECT		<input type="checkbox"/>
SAP_SCMA_TEST_RUN		<input type="checkbox"/>
SAP_SCMA_VALUE_DATE		<input type="checkbox"/>

Important Note: Value for the source ID should match the value of the source system created using **Configuration Cockpit** application as described [here](#).

The source ID defined in this section it's not valid for handling units replication also. The source id for Handling Units replication it is specified in the selection parameters defined in the report. For more details, please read [handling units replication report selection parameters](#) details.

Create and activate Change Pointers

Through Change Pointer's mechanism if any master data field value is changed in SAP like Material master, Vendor master etc. through standard t-code like MM01, MM02, VA01, VA02, the corresponding records will be created in change pointers tables (BDCP, BDCPS).

RDP Integration package installation does not create the change pointers automatically. Change pointers needs to be created and activated manually.

Services using change pointers mechanism:

- Batch
- Customer
- Supplier
- Product

By default, RDP Integration reports are using specific change pointers listed below. These separate message types below need to be created and it's not recommended to re-use existing message types used in other processes also:

- /VPCOE/BATCH
- /VPCOE/CUSTOMER
- /VPCOE/SUPPLIER
- /VPCOE/PRODUCT

Also please make sure the minimum change items described below are assigned to the corresponding message types.

Follow the steps below to create the change pointers for each of the services and activate them:

1. Create message type (tcode WE81)

Message Type	Short Text
/VPCOE/BATCH	Batch
/VPCOE/CUSTOMER	Customer
/VPCOE/PRODUCT	Product
/VPCOE/SUPPLIER	Supplier

2. Assign change items to message type (tcode BD52)

Assign the table fields to message types which you want to be considered for change pointers generations.

Following are the minimum fields to be assigned for each message type:

Message type:	Object	Table name	Filed name
/VPCOE/BATCH	CHARGE	MCHA	CHARG
	CHARGE	MCHA	HSDAT
	CHARGE	MCHA	KEY
		MCHA	WERKS
	ESSUB	ESTRH	SUBID
	MATERIAL	MARA	MATNR
/VPCOE/PRODUCT	MATERIAL	DMAKT	MAKTX
	MATERIAL	DMARM	KEY
	MATERIAL	MARA	KEY
	MATERIAL	MARA	LVORM
/VPCOE/CUSTOMER	DEBI	KNA1	KEY
	DEBI	KNA1	LOEVM
	DEBI	KNA1	NAME1
	DEBI	KNA1	NAME2
	DEBI	KNA1	NODEL
/VPCOE/SUPPLIER	KRED	LFA1	KEY
	KRED	LFA1	LOEVM
	KRED	LFA1	NAME1
	KRED	LFA1	NODEL
	KRED	LFB1	KEY

3. Activate change pointers generally (tcode BD61)

4. Activate change pointers corresponding to fields in message types (transaction code BD50).

Set Active flag for each of the message type created and Save it.

5. Also check if the data elements of all the required fields have 'change Document' field checked (Below screen shot for Material Number)

Dictionary: Display Data Element

The screenshot shows the SAP Dictionary: Display Data Element interface. At the top, there is a toolbar with various icons. Below it, the 'Data element' is set to 'MATNR' and the 'Short Description' is 'Material Number'. A blue bar at the bottom contains tabs: 'Attributes' (selected), 'Data Type', 'Further Characteristics', and 'Field Label'. In the 'Attributes' section, there is a 'Search Help' field. Under 'Name', the value is 'S_MAT1'. Under 'Parameters', the value is 'MATNR'. Below these, 'Parameter ID' is 'MAT'. Under 'Default Component Name', the value is 'MATERIAL'. At the bottom of this section, there is a checkbox labeled 'Change document' which is checked and highlighted with a red box.

CONTENT OF RDP INTEGRATION PACKAGE

Once you have done all the above steps successfully, your RDP Integration Accelerator package is ready to use.

Report list and assigned transaction codes included in the integration accelerator package:

Development Package	Report Name	Transaction Code
/VPCOE/COMMON_INTEGRATION	/VPCOE/REPROCESS_PAYLOAD	/VPCOE/REPROCESS
/VPCOE/RDP_INTEGRATION	/VPCOE/BATCH_TO_RDP	/VPCOE/RDP_BATCH
	/VPCOE/CONFIGURATION_TO_RDP	/VPCOE/RDP_CONF
	/VPCOE/ORG_DATA_TO_RDP	/VPCOE/RDP_ORG_DATA
	/VPCOE/PRODUCT_TO_RDP	/VPCOE/RDP_PRODUCT
	/VPCOE/CUSTOMER_TO_RDP	/VPCOE/RDP_CUSTOMER
	/VPCOE/SUPPLIER_TO_RDP	/VPCOE/RDP_SUPPLIER
	/VPCOE/DELIVERY_TO_RDP	/VPCOE/RDP_DELIVERY
	/VPCOE/MAT_DOC_TO_RDP	/VPCOE/RDP_MAT_DOC
	/VPCOE/BILLDOCITEMS_TO_RDP	-
	/VPCOE/SUPPLINVITEMS_TO_RDP	-
	/VPCOE/INVENTORY_TO_RDP	/VPCOE/RDP_INVENTORY
	/VPCOE/INCOTERMS_TO_RDP	/VPCOE/RDP_INCOTERMS
	/VPCOE/TO_RDP	/VPCOE/RDP
/VPCOE/PLM_INTEGRATION	/VPCOE/R_UPH_PCKG_CMP_PLM_LOAD	/VPCOE/UPE_PLM
	/VPCOE/R_UPH_PCKG_CMP_RCP_LOAD	/VPCOE/UPE_RCP
	/VPCOE/R_UPH_PCKG_DATA_LOAD	/VPCOE/UPE_PLM
/VPCOE/OBJ_INTEGRATION	/VPCOE/R_UPH_PCKG_MCL_LOAD	/VPCOE/UPE_MCL
	/VPCOE/R_UPH_PCKG_CMP_BOM_LOAD	/VPCOE/UPE_BOM
	/VPCOE/R_UPH_HU_DATA_LOAD	/VPCOE/UPE_PCI_HU
/VPCOE/RDP_PCKGFEE_INTEGRATION	/VPCOE/R_PCKF_POSTING	/VPCOE/PCKF_POSTING
	/VPCOE/R_PCKF_RETRIEVAL	/VPCOE/PCKF_RETRIEVE

	/VPCOE/R_PCKF_CUSTOMER_ROLE	/VPCOE/PCKF_CUS_ROLE
	/VPCOE/R_PCKF_CUSTOM_EXEMPT	/VPCOE/PCKF_CUS_EXMP
	/VPCOE/R_PCKF_COMP_CODE_EXEMPT	/VPCOE/PCKF_CC_EXMP

Transaction code /VPCOE/RDP has been created to run the general report /VPCOE/TO_RDP. If it's necessary, please create your own custom role with following authorizations and assign it to specific users to allow them to execute the transaction.

Authorization needed to execute TCode /VPCOE/RDP (in foreground mode):

Object	Field Name	From	To Activity
S_TCODE	TCD	/VPCOE/RDP	
S_ADMIN_FCD	S_ADMIN_FCD		STOM
S_GUI	ACTVT		61
S_ALV_LAYO	ACTVT		23
W_BETR_USR	ACTVT		03
W_BETR_USR	WERKS	*	
M_MSEG_WMB	ACTVT		03
M_MSEG_WMB	WERKS	*	
M_MATE_MAR	ACTVT		03
M_MATE_MAR	BEGRU		
M_MATE_MAT	ACTVT		03
M_MATE_MAT	BEGRU		
M_MATE_WGR	ACTVT		03
M_MATE_WGR	BEGRU		
K_KEKO	ACTVT		03
K_KEKO	BUKRS	*	
K_KEKO	KLVAR		

Please note that exact authorization setup can vary from customer to customer, and security team can determine and setup the roles with all authorizations required.

Authorization needed to run the reports in background mode is needed for the transaction code for each specific report.

HOW TO REPLICATE DATA TO RDP

RDP Integration reports will extract needed data from your system, pack the data in the required format, and send them to RDP.

RDP Integration interfaces will support two scenarios to load the data (excepting organization and configuration data which only supports Initial load):

1. **Initial Load** – In this case, all the objects selected based on filters criteria will be sent to RDP
2. **Delta Load** – In this case, all changed (new, changed) business object instances from the last load will be sent to RDP.

The below steps describe how you can use the integration accelerator to load data into the RDP.

RDP ERP Integration reports

Each report in the package, for each service id, can be executed individually using transaction code SE38 or in background mode.

For a proper replication we propose to follow the extraction of the data in the following order for an initial replication:

Initial data replication - sequence



For delta load replication, you can consider daily or monthly replication or depending on your needs:

Delta data replication - sequence



Also, one general report for RDP ERP is available, /VPCOE/TO_RDP which helps you to load the data to RDP for all services available from one single place. Transaction code /VPCOE/RDP has been created to execute /VPCOE/TO_RDP report.

General Approach for Loading Data

The RDP Integration Accelerator provides multiple options for loading data into **SAP Responsible Design and Production (RDP)**. The approach varies depending on the type of data being processed and the desired load method.

Initial Load

- All objects selected based on the defined filter criteria are sent to RDP.
- For all entities, a new import overwrites and deletes the existing data in RDP.

Get Only Delta Flag

- Extracts delta data based on **change pointers** or **specific report logic**.
- Only newly created or modified objects are selected and sent to RDP.

Preview Mode

- Displays the data that would be sent to RDP based on the selection criteria without transferring the data.
- The results are shown on the screen after executing the report.

Test Mode

- **Available for packaging data reports.**
- When the **Test Mode** checkbox is selected, the data extraction and mapping processes remain the same, but the transfer of data to the RDP API is skipped.
- Test Mode is supported in both **initial load** and **delta load** scenarios.

Export to XLSX File

- Allows the generation of an Excel file based on the selected criteria.
- The exported data can be used with the **Data Import** application in RDP.
- Note: The number of records exported to the XLSX file is limited to **50,000**.

Selection ID

- Whenever an **initial load** or **delta load** is successfully performed, the selection parameters along with the current date are saved.
- This saved **Selection ID** is used to determine the next delta load.
- The last usage date of the Selection ID is also recorded for reference.

Reports with Specific Load Functionality

The following reports do not support **delta load** functionality. If data changes occur, a re-execution of the **initial load** is required:

- **/VPCOE/CONFIGURATION_TO_RDP** – Configuration data replication.
- **/VPCOE/ORG_DATA_TO_RDP** – Organization data replication.
- **/VPCOE/INCOTERMS_TO_RDP** – Incoterms data replication.

Reports Using Change Pointers for Delta Load

The following reports use **change pointers** to identify and replicate all new or changed objects since the last replication:

- **/VPCOE/BATCH_TO_RDP** – Batch data replication.
- **/VPCOE/PRODUCT_TO_RDP** – Product data replication.
- **/VPCOE/CUSTOMER_TO_RDP** – Customer data replication.
- **/VPCOE/SUPPLIER_TO_RDP** – Supplier data replication.

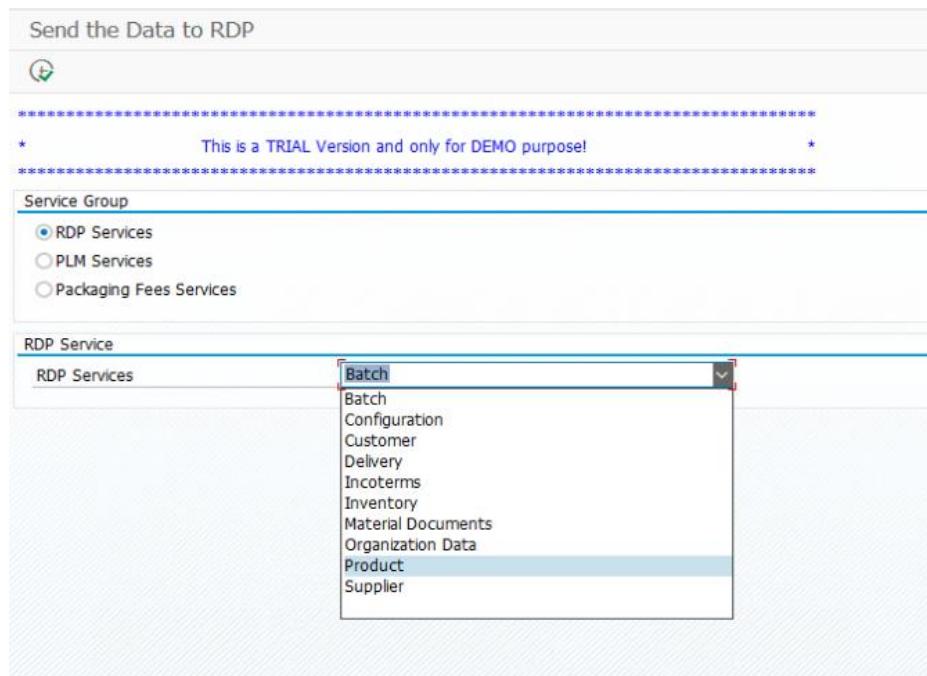
Reports with Delta Load Logic

The following reports do not rely on change pointers and instead apply their own logic to determine delta data:

- **/VPCOE/DELIVERY_TO_RDP** – Delivery data replication.
- **/VPCOE/MAT_DOC_TO_RDP** – Material documents replication.
- **/VPCOE/INVENTORY_TO_RDP** – Inventory data replication.

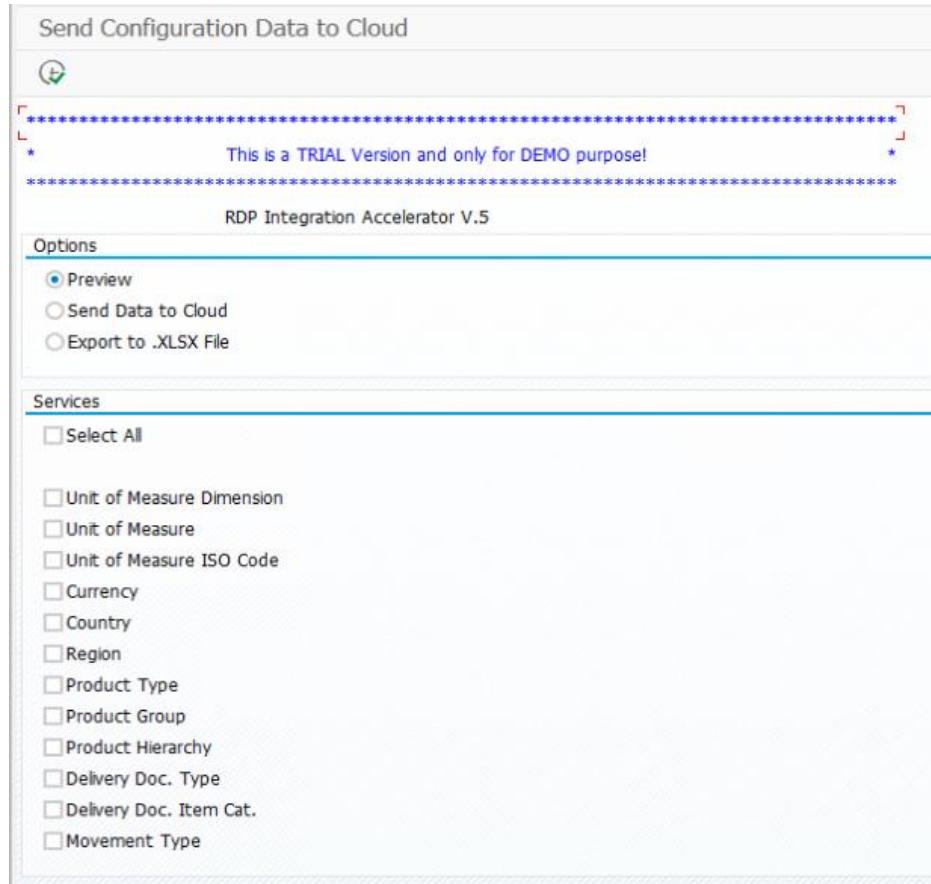
Open transaction code /VPCOE/RDP.

Select necessary service group and service name and execute the report:



Loading Configuration data

Open transaction code VPCOE/RDP and select service ID 'Configuration' (alternatively you can run the report /VPCOE/CONFIGURATION_TO_RDP):



Executing the Configuration report you can load specific configuration data related to:

- Unit of Measure
- Unit of Measure Dimension
- Unit of Measure ISO Code
- Currency
- Country
- Region
- Product Type
- Product Group
- Product Hierarchy
- Delivery Document Type
- Delivery Document Item Category
- Movement Type

Note: You can select all the services or only needed ones in order to send the data to RDP.

Note: Internal units of measure will be sent to RDP.

Loading Incoterms

Open transaction code VPCOE/RDP and select service ID 'Incoterms' (alternatively you can run the report /VPCOE/INCOTERMS_TO_RDP):

Send Incoterms to Cloud

This is a TRIAL Version and only for DEMO purpose!

RDP Integration Accelerator V.5

Options

- Preview
- Send Data To Cloud
- Export to .XLSX File

Loading Organization data

Open transaction code VPCOE/RDP and select service 'Organization data' (alternatively you can run the report /VPCOE/ORG_DATA_TO_RDP):

Send Organization Data to RDP

This is a TRIAL Version and only for DEMO purpose!

RDP Integration Accelerator V.5

Options

- Preview
- Send Data to Cloud
- Export to .XLSX File

Selection Parameter

Select All

Company Code
 Plant
 Sales Organization
 Distribution Channel
 Division
 Sales Area

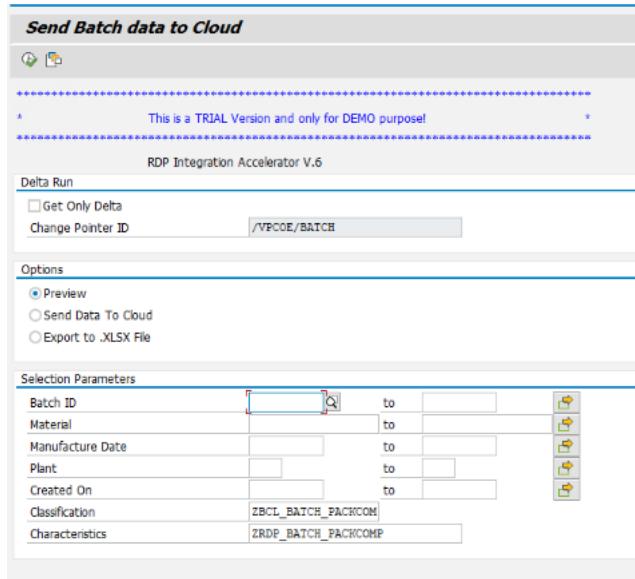
Executing the Organization data report you can load specific organization data related to:

- Company Code
- Plant
- Sales Organization
- Distribution Channel
- Division
- Sales Area

Note: You can select all the services or only needed ones in order to send the data to RDP.

Loading Batch data

Open transaction code VPCOE/RDP and select the service 'Batch data' (alternatively you can run the report /VPCOE/BATCH_TO_RDP):



Initial Load Approach

The initial load extracts and transfers all relevant batch data based on the defined selection criteria. The process includes:

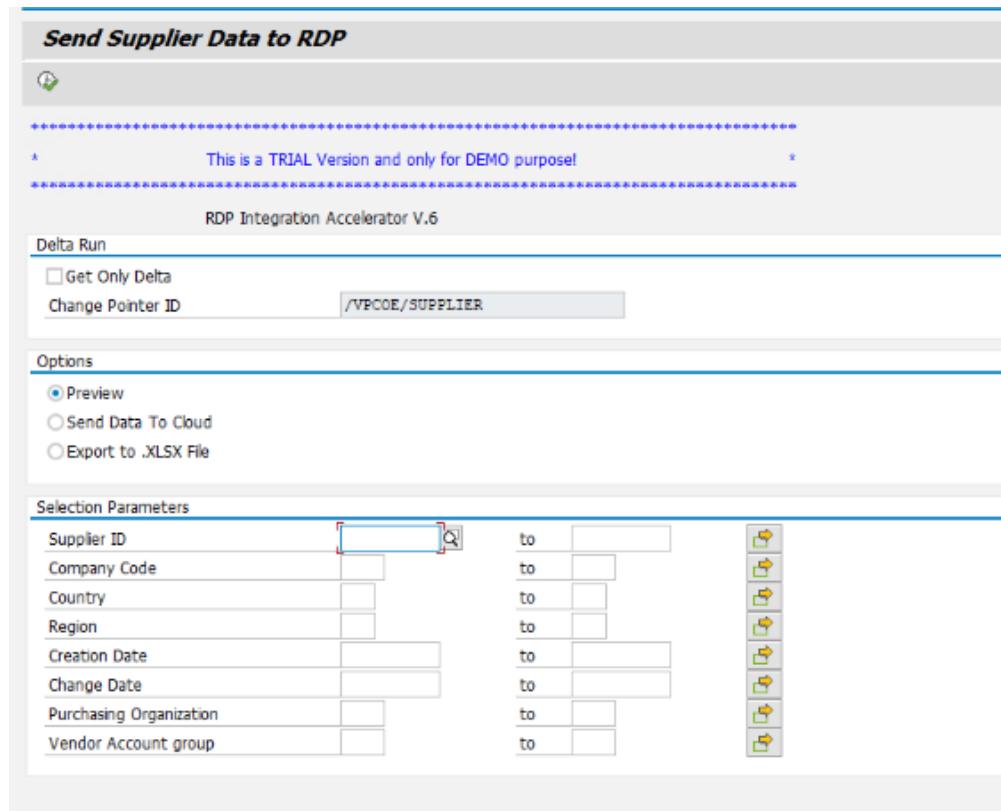
- Selection of all batch data based on specified filters.
- Determination of Packaging Composition in Batch:
 - The report identifies all batches matching the existing filter criteria for classes and characteristics.
 - If a batch contains a characteristic entry related to Packaging Composition ID, the corresponding characteristic value is retrieved.
 - For batches with an entry in classes and characteristics, the Packaging Composition ID is additionally transferred to RDP.
 - The identified Packaging Composition is additionally replicated.

Delta Load Approach

- The Delta Load functionality operates based on Change Pointers, ensuring that only newly created or modified batch data is selected.
- The system identifies changes using the defined change pointer logic and replicates the relevant updates to RDP.

Loading Supplier data

Open transaction code VPCOE/RDP and select service 'Supplier' (alternatively you can run the report /VPCOE/SUPPLIER_TO_RDP):



Initial Load Approach

During the initial load, all relevant supplier data is extracted and transferred to RDP based on the specified selection criteria:

- All suppliers matching the defined filters are selected and sent to RDP.
- Suppliers **marked as deleted** are also included in the replication to ensure data consistency. This is a requirement/recommendation on RDP.

Delta Load Approach

- The **Delta Load** functionality is based on **Change Pointers**, ensuring that only newly created, modified, or deleted suppliers are selected for replication.
- All selection parameters remain available in delta load mode, allowing users to **restrict the scope of data replication** as needed.

Tax Number and Tax Code Selection Logic for Supplier Data

During supplier (vendor) replication, tax number and tax code are selected and transferred to RDP. The selection logic depends on the system configuration and whether **Customer Vendor Integration (CVI)** is active:

Logic to Select Tax Number:

- **If CVI is active:**
 - Tax number is selected from:
 - **DFKKBPTAXNUM-TAXNUM** or
 - **DFKKBPTAXNUM-TAXNUMXL**
- **If CVI is not active:**
 - Tax number is selected from:
 - **LFA1-STCEG** or
 - **LFAS-STCEG**

Note:

- The usage of **VAT registration codes and tax numbers** in customer/vendor master data may vary depending on country-specific tax regulations and the organization's specific requirements.
- If the default selection logic does not meet the organization's requirements, a **BAdI implementation** is available to **overwrite the logic** for tax number selection.

Loading Customer and Customer Extension data

Open transaction code VPCOE/RDP and select service 'Customer' (alternatively you can run the report /VPCOE/CUSTOMER_TO_RDP):

The screenshot shows the 'Send Customer Data to Cloud' interface. At the top, there is a note: 'This is a TRIAL Version and only for DEMO purpose!'. Below that, it says 'RDP Integration Accelerator V.6'. The 'Delta Run' section contains a checkbox for 'Get Only Delta' and a field for 'Change Pointer ID' set to '/VPCOE/CUSTOMER'. The 'Options' section has a radio button for 'Preview' selected, and other options like 'Send Data to Cloud' and 'Export to .XLSX File'. The 'Type of API to be triggered' section has checkboxes for 'Common API' (checked) and 'Extension API' (checked). The 'Selection Parameters' section lists fields for Customer ID, Company Code, Country, Region, Creation Date, Change Date, Sales Organization, and Customer Account group, each with a search icon and a 'to' field. To the right of these fields is a vertical stack of icons. The 'Customer Extension Selection Criteria' section includes a 'Customer Role' field and a 'to' field, followed by another stack of icons.

You can choose between two types of APIs based on the required data:

- **Common API** – Select this checkbox to send **customer data** based on the defined selection criteria to RDP.
- **Extension API** – Select this checkbox to send **additional attributes** based on the selection criteria, as implemented in the BAdI.
 - For details on extending functionality, refer to the [Extensibility](#) section.

Initial Load Approach

During the initial load, all relevant customer data is extracted and transferred to RDP:

- All customers matching the defined filters are selected and sent to RDP.

- Customers **marked as deleted** are also included in the replication to ensure data consistency.

Delta Load Approach

- The **Delta Load** functionality is driven by **Change Pointers**, ensuring that only newly created, modified, or deleted customer records are selected.
- Selection can be further refined using selection filters to optimize data extraction.
- All selection fields are available in delta load mode, allowing users to **restrict data selection** as needed.

Loading Customer extension data

- To load **Customer Extension Data**, select the **Extension API** checkbox.
- For further information, refer to the [Extensibility capabilities for RDP Integration reports](#).

Sample BAdI Implementation for Customer Extensions

The sample BAdI implementation follows RDP recommendations for defining classes and characteristics as follows:

Class	ZCCL_RDP_CUST_EXT
Description	RDP specific Customer Attributes
Class Type	011

Characteristics	Description	Data Type	Length	Decimal	UoM / Values
ZRDP_CUST_EXT_ROLE	Customer Role	CHAR	10	0	Code List
ZRDP_PROD_EXT_TAG	Customer Tag	CHAR	10	0	Customer defined code list

Tax Number and Tax Code Selection Logic

During customer replication, tax number and tax code are selected and transferred to RDP. The selection logic depends on the system configuration:

- If CVI is active** in ECC (and in S/4H):
 - Tax number is selected from **DFKKBPTAXNUM-TAXNUM** or **DFKKBPTAXNUM-TAXNUMXL**.
- If CVI is not active**:
 - Tax number is selected from **KNA1-STCEG** or **KNAS-STCEG**.

Note:

The usage of VAT registration codes and tax numbers in customer/vendor master data can vary depending on country-specific tax regulations and organizational requirements.

If customization is required, use the **available BAdI** to overwrite the default logic for tax number selection.

Please use the BAdI available if you need to overwrite the above logic for tax number information based on your specific needs.

Loading Product and Product extension data

Open transaction code VPCOE/RDP and select service 'Product' (alternatively you can run the report /VPCOE/PRODUCT_TO_RDP):

Send Product Master Data to RDP

RDP Integration Accelerator V.6

Delta Run

Get Only Delta
Change Pointer ID

Options

Preview
 Send Data to Cloud
 Export to .XLSX File

Type of API to be triggered

Common API
 Extension API

Product Selection Criteria

Material	<input type="text"/>	<input type="button" value="Search"/>	<input type="button" value="New"/>
Material Type	<input type="text"/>	<input type="button" value="to"/>	<input type="button" value="New"/>
Material Group	<input type="text"/>	<input type="button" value="to"/>	<input type="button" value="New"/>
Last Change	<input type="text"/>	<input type="button" value="to"/>	<input type="button" value="New"/>
Sales Organization	<input type="text"/>	<input type="button" value="to"/>	<input type="button" value="New"/>
Plant	<input type="text"/>	<input type="button" value="to"/>	<input type="button" value="New"/>
Plant Status	<input type="text"/>	<input type="button" value="to"/>	<input type="button" value="New"/>

Product Extension Selection Criteria

Product Content	<input type="text"/>	<input type="button" value="to"/>	<input type="button" value="New"/>
Product Type Harmonized	<input type="text"/>	<input type="button" value="to"/>	<input type="button" value="New"/>
Valid From	<input type="text"/>	<input type="button" value="to"/>	<input type="button" value="New"/>

You can choose between two types of APIs based on the required data:

- **Common API** – Select this checkbox to send **product data** based on the defined selection criteria to RDP.
- **Extension API** – Select this checkbox to send **additional attributes** based on the selection criteria, as implemented in the BAdI.
 - For more information on extending functionality, refer to the [Extensibility section](#)

Initial Load Approach

During the initial load, all relevant product data is extracted and transferred to RDP:

- All products matching the defined filters are selected and sent to RDP.

Delta Load Approach

- The **Delta Load** functionality is based on **Change Pointers**, ensuring that only newly created, modified, or deleted product records are selected.
- The system identifies changes using the change pointers mentioned on the selection screen and applies the **Last Change Date** filters defined in the **Product Selection Criteria**.
- **All products, including deleted products**, are included in the replication to ensure data consistency.

Loading Product extension data

- to load **Product Extension Data**, select the **Extension API** checkbox.
- For further information, refer to the [Extensibility capabilities for RDP Integration reports](#).

Loading Deliveries

Open transaction code VPCOE/RDP and select service 'Delivery' (alternatively you can run the report /VPCOE/DELIVERY_TO_RDP):

Initial load approach:

- all the deliveries selected based on filters criteria will be sent to RDP. Please note only deliveries with status 'Completely Process' are considering for replication.
- by selecting 'Send related customer before' checkbox, all the related customers according to selection criteria and for the change pointers specified will be also sent to RDP
- all deleted deliveries will be sent to RDP

Delta load approach for Delivery documents report:

- All the objects selected from last successful replication, will be sent to RDP. Please note only deliveries with status 'Completely Process' are considering for delta replication.
- Some filter criteria become unchangeable, to assure delta load consistency for all filters combinations used before

When an initial load is successfully executed, the selection parameters used for initial load are saved into so called 'Selection ID' together with selection parameters and creation date/change date.

- The **Delivery Document Created Date/Changed Date** is predefined programmatically in this case. These values contain the date since the next Delta can be retrieved up to current

date. In this way, all the new created/posted documents after last successful replication till current date will be selected for delta load.

Send Delivery to RDP

* This is a TRIAL Version and only for DEMO purpose! *

RDP Integration Accelerator V.5

Delta Run

<input checked="" type="checkbox"/> Get Only Delta	DLVR-Delivery-20220126	Delete Current Selection ID
Previous Selection Id		Delete all Selection IDs

Options

Send Related Customers before
 Change Pointer ID for Customer /VPCOE/CUSTOMER
 Preview
 Send Data to Cloud
 Export to .XLSX File

Selection Parameters

Delivery Type	to	
Sales Organization	to	
Ship-to Party	to	
Ship-to Country	to	
Region	to	
Item Category	to	
Distribution Channel	to	
Division	to	
Delivery Number	to	
Actual Goods Movement Date	to	
Delivery Plant Country	to	

Selection Dates

Created on	26.01.2022	to	25.01.2024	
OR				
Changed on	26.01.2022	to	25.01.2024	

Sold-to party to

Loading Material documents

Open transaction code VPCOE/RDP and select service ‘Material Documents’ (alternatively you can run the report /VPCOE/MAT_DOC_TO_RDP):

Send Material Documents to RDP

This is a TRIAL Version and only for DEMO purpose!

RDP Integration Accelerator V.5

Variants

Get Only Delta

Options

Selection Id: DOC-Material Documents-20240627

Preview
 Send Data to Cloud
 Export to .XLSX File

Billing Document Items

Get Billing Document Items

Supplier Invoice Items

Get Supplier Invoice Items

Selection Parameters

Country	<input type="text" value="DE"/> <input type="button" value="Q"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>
Posting Date	<input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>
Material Document Year	<input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>
Plant	<input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>
Material Document	<input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>
Movement Type	<input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>
Material	<input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>	<input type="button" value="to"/> <input type="text"/>

Variants

- Goods Receipt with Production Order reference in specific country
- Goods Receipt with Stock Transfer reference imported in specific country
- Goods Receipt with Stock Transfer reference in specific country (domestic)
- Goods Issue with Stock Transfer reference exported from specific country
- Goods Issue with Stock Transfer reference in specific country (domestic)
- Goods Receipt with Purchase order reference imported in specific country
- Goods Receipt with Purchase order reference in specific country (domestic)

All the goods movements according to selection criteria will be selected. Country is a mandatory selection filter. Selected country can represent Ship From and/or Ship To country, depending on the variant selected.

Group 1: Goods Receipt with Production Order reference in specific country

All goods receipts will be replicated with manufacturing order reference, but no supplier and customer reference and not from a stock transfer.

Group 2: Goods Receipt with Stock Transfer reference imported in specific country

All goods receipt with stock transfer reference imported in selected country will be replicated.

Group 2b: Goods Receipt with Stock Transfer reference in specific country (domestic)

All goods receipt with stock transfer reference within same country will be replicated.

Group 3: Goods Issue with Stock Transfer reference exported from specific country

All goods issue with stock transfer reference exported from selected country will be replicated.

Group3b: Goods Issue with Stock Transfer reference in specific country (domestic)

All goods issue with stock transfer reference within same country will be replicated.

Group 4: Goods Receipt with Purchase order reference imported in specific country

All goods receipt with purchased order reference imported in specific country and not from a stock transfer.

Group 5: Goods Receipt with Purchase order reference in specific country (domestic)

All goods receipt with reference to Purchase order within the same country (Domestic procurement).

Initial load approach for Material documents report:

- All the objects selected based on filters criteria and for variant(s) selected will be sent to RDP
- If the **Billing Documents** and/or **Invoice Documents** checkboxes are selected, the system retrieves the respective documents based on the defined selection criteria.
- **Country Restriction:** Billing documents and invoice item documents are **selected only for Spain country**.

Send Material Documents to RDP

RDP Integration Accelerator V.6

Variants

Get Only Delta

Options

Selection Id: DOC-Material Documents-20250313

Preview
 Send Data to Cloud
 Export to .XLSX File

Billing Document Items

Get Billing Document Items

Billing Document Country	ES	to	<input type="button" value="..."/>
Billing Document	<input type="text"/>	to	<input type="button" value="..."/>
Billing Document Date	<input type="text"/>	to	<input type="button" value="..."/>
SD Document Category	<input type="text"/>	to	<input type="button" value="..."/>

Supplier Invoice Items

Get Supplier Invoice Items

Supplier Invoice Country	ES	to	<input type="button" value="..."/>
Supplier Invoice	<input type="text"/>	to	<input type="button" value="..."/>
Supplier Invoice Item	<input type="text"/>	to	<input type="button" value="..."/>
Company Code	<input type="text"/>	to	<input type="button" value="..."/>
Supplier Invoice Plant	<input type="text"/>	to	<input type="button" value="..."/>
Fiscal Year	<input type="text"/>	to	<input type="button" value="..."/>

Selection Parameters

Country	<input type="text"/> <input type="button" value="..."/> <input type="button" value="Q"/>	to	<input type="button" value="..."/>
Posting Date	<input type="text"/>	to	<input type="button" value="..."/>
Material Document Year	<input type="text"/>	to	<input type="button" value="..."/>
Plant	<input type="text"/>	to	<input type="button" value="..."/>
Material Document	<input type="text"/>	to	<input type="button" value="..."/>
Movement Type	<input type="text"/>	to	<input type="button" value="..."/>
Material	<input type="text"/>	to	<input type="button" value="..."/>

Variants

Goods Receipt with Production Order reference in specific country
 Goods Receipt with Stock Transfer reference imported in specific country
 Goods Receipt with Stock Transfer reference in specific country (domestic)
 Goods Issue with Stock Transfer reference exported from specific country
 Goods Issue with Stock Transfer reference in specific country (domestic)
 Goods Receipt with Purchase order reference imported in specific country
 Goods Receipt with Purchase order reference in specific country (domestic)

Delta load approach for Material documents report:

- All the objects selected (based on posting date) from last load, will be sent to RDP
 - Some filter criteria become unchangeable, to assure delta load consistency for all filters combinations used before
- When an initial load is successfully executed, the selection parameters used for initial load are saved into so called 'Selection ID' together with selection parameters and creation date/posting date.
- The **Material Document Posting Date** is predefined programmatically in this case. These values contain the date since the next Delta can be retrieved up to current date. In this way, all the new posted documents after last successful replication till current date will be selected for delta load.

Send Material Documents to RDP

* This is a TRIAL Version and only for DEMO purpose! *

RDP Integration Accelerator V.5

Variants

<input checked="" type="checkbox"/> Get Only Delta	Previous Selection Id	DOC-Material Documents-20220913	Delete Current Selection ID
			Delete all Selection IDs

Options

Preview

Send Data to Cloud

Export to .XLSX File

Selection Parameters

Country	US	to		<input type="button" value="New"/> <input type="button" value="Open"/> <input type="button" value="Save"/> <input type="button" value="Print"/> <input type="button" value="Delete"/>
Posting Date	15.01.2024	to	25.01.2024	
Material Document Year		to		
Plant		to		
Material Document		to		
Movement Type		to		

Variants

Goods Receipt with Production Order reference in specific country

Goods Receipt with Stock Transfer reference imported in specific country

Goods Receipt with Stock Transfer reference in specific country (domestic)

Goods Issue with Stock Transfer reference exported from specific country

Goods Issue with Stock Transfer reference in specific country (domestic)

Goods Receipt with Purchase order reference imported in specific country

Goods Receipt with Purchase order reference in specific country (domestic)

Important Notes for Material Documents Loading

- **Stock Type and Specialized Stock Type:**
Goods movement data are not restricted based on stock type or specialized stock type. All relevant data is extracted and transferred without applying stock-specific restrictions.
- **Emergency Corrections:**
If an emergency correction is performed after data extraction, it is recommended to re-execute the initial load to ensure that corrected data is transferred to RDP.
- **Data Overwrite During Import:**
For all material documents, a new import will overwrite and delete the existing data in RDP. User needs to ensure that the most accurate data is reloaded when necessary.

- **Handling Movement Types 301, 303, 305 in Material Documents**

During the loading of **material documents**, the Integration Accelerator retrieves relevant data from the source system and processes it according to predefined conditions. Standard filters and mapping rules are applied to ensure accurate and consistent data replication.

Default Behavior for Movement Types

- By default, the Integration Accelerator does not use movement types as selection criteria.
- However, **movement types 301, 303, and 305** are included in the standard logic to ensure correct data selection and replication.
- If a customer has defined **custom movement types (Z-types)** corresponding to any of these standard movement types (**301, 303, or 305**), they must be explicitly specified in the BAdI implementation using the method:

```
/VPCOE/IF_ADJ_DATA_RETRIEVAL~DEFINE_MOVEMENT_TYPE
```

When to Use This Method:

- **Custom Movement Types Exist** – If custom movement types replace standard movement types (**301, 303, or 305**) and need to be considered during data selection.
- **Ensuring Correct Data Selection** – To apply the appropriate selection conditions for these custom movement types.

No Implementation of this method is required if:

- No custom movement types exist for **301, 303, and 305**.

For more details on badi implementation example please have a look in the method itself of sample class implementation and [here](#).

Loading Billing Document Items and Supplier Invoice Items

For loading billing document items and supplier invoice items, the existing reports for material documents and delivery replication should be used.

User needs to select the respective checkboxes available to select and send billing documents and/or invoice document items for respective material documents selected. Only the matching objects will be replicated.

Material documents:

Send Material Documents to RDP

This is a TRIAL Version and only for DEMO purpose!

RDP Integration Accelerator V.5

Variants

Get Only Delta

Options

Selection Id: DOC-Material Documents-20240627

Preview
 Send Data to Cloud
 Export to .XLSX File

Billing Document Items

Get Billing Document Items

Billing Document Country	ES	to	<input type="checkbox"/>	
Billing Document	<input type="text"/>	to	<input type="checkbox"/>	
Billing Document Date	<input type="text"/>	to	<input type="checkbox"/>	
SD Document Category	<input type="text"/>	to	<input type="checkbox"/>	

Supplier Invoice Items

Get Supplier Invoice Items

Supplier Invoice Country	ES	to	<input type="checkbox"/>	
Supplier Invoice	<input type="text"/>	to	<input type="checkbox"/>	
Supplier Invoice Item	<input type="text"/>	to	<input type="checkbox"/>	
Company Code	<input type="text"/>	to	<input type="checkbox"/>	
Supplier Invoice Plant	<input type="text"/>	to	<input type="checkbox"/>	
Fiscal Year	<input type="text"/>	to	<input type="checkbox"/>	

Selection Parameters

Country	<input type="text"/> @ ... *	<input type="button" value="..."/>	<input type="checkbox"/>		
Posting Date	20.02.2024	to	27.06.2024	<input type="checkbox"/>	
Material Document Year	<input type="text"/>	to	<input type="checkbox"/>		
Plant	<input type="text"/>	to	<input type="checkbox"/>		
Material Document	<input type="text"/>	to	<input type="checkbox"/>		
Movement Type	<input type="text"/>	to	<input type="checkbox"/>		
Material	<input type="text"/>	to	<input type="checkbox"/>		

Variants

- Goods Receipt with Production Order reference in specific country
- Goods Receipt with Stock Transfer reference imported in specific country
- Goods Receipt with Stock Transfer reference in specific country (domestic)
- Goods Issue with Stock Transfer reference exported from specific country
- Goods Issue with Stock Transfer reference in specific country (domestic)
- Goods Receipt with Purchase order reference imported in specific country
- Goods Receipt with Purchase order reference in specific country (domestic)

Deliveries:

Send Delivery to RDP

This is a TRIAL Version and only for DEMO purpose!

RDP Integration Accelerator V.5

Delta Run

Get Only Delta

Options

Send Related Customers before

Selection Id: DLVR-Delivery-20240627

Preview

Send Data to Cloud

Export to .XLSX File

Get Billing Document Items

Billing Document Country	ES	to	<input type="checkbox"/>
Billing Document		to	<input type="checkbox"/>
Billing Document Date		to	<input type="checkbox"/>
SD Document Category		to	<input type="checkbox"/>

Selection Parameters

Delivery Type		to	<input type="checkbox"/>
Sales Organization		to	<input type="checkbox"/>
Ship-to Party		to	<input type="checkbox"/>
Ship-to Country		<input checked="" type="checkbox"/>	<input type="checkbox"/>
Region		to	<input type="checkbox"/>
Item Category		to	<input type="checkbox"/>
Distribution Channel		to	<input type="checkbox"/>
Division		to	<input type="checkbox"/>
Delivery Number		to	<input type="checkbox"/>
Actual Goods Movement Date		to	<input type="checkbox"/>
Delivery Plant Country		to	<input type="checkbox"/>
Material		to	<input type="checkbox"/>

Selection Dates

Created on		to	<input type="checkbox"/>
OR			
Changed on		to	<input type="checkbox"/>

This replication for additional Billing documents/Supplier Invoices is required **only for Spanish customers** therefore replication for billing documents/invoices are restricted to only documents related to Spain (therefore Billing Document Country and Supplier Invoice Country pre-defaulted with 'ES' value) and only for documents with status Posted.

Supplier invoices and billing documents information for the transactional documents will be selected accordingly as below:

- A reference to **Supplier Invoice** with link to Material document for:
 - *Goods Receipts with Purchase Order reference imported in Specific country where*
 - Country <> ES and ShipToCountry = ES
 - *Goods Receipts with Purchase Order reference in specific country (domestic) where*
 - ShipFromCountry = ES and ShipToCountry = ES
- A reference to **Billing Document** with link to material document for:
 - *Goods Receipts with Stock Transfer reference imported in specific country where*
 - ShipFromCountry <> ES and ShipToCountry = ES

- ReferenceSDDocumentCategory = Delivery (J) , Reversal (T)
 - billings document posting status VBRK-RFBSK = C, H (C-“Posting document has been created”; H- “Posted via invoice list”)
- *Goods Issue with Stock Transfer exported from Specific country where*
 - ShipFromCountry = ES and ShipToCountry <> ES
 - ReferenceSDDocumentCategory = Delivery (J) , Reversal (T)
 - billings document posting status VBRK-RFBSK = C, H (C-“Posting document has been created”; H- “Posted via invoice list”)
- For domestic stock transfer - no billing documents is required for
 - *Goods Receipts with Stock Transfer reference in specific country (domestic)*
 - *Goods Issues with Stock Transfer reference in specific country (domestic)*
- For deliveries as exports
 - A reference to billing documents with link to Material document where
 - ShipFromCountry = ES and ShipToCountry <> ES
 - ReferenceSDDocumentCategory = C, G, H, I, K, L, J, T
 - billings document posting status VBRK-RFBSK = C, H (C - “Posting document has been created”; H- “Posted via invoice list”)
- Cancellations are excluded from reporting therefore no reference documents need to be sent for these.
- Returns are not excluded from reporting, therefore reference to account documents are replicated for:
 - Material documents with PO return - domestic
 - Material documents with PO return - not domestic
 - Material documents with Stock Transfer return - not domestic
- No replication of Supplier Invoices/Billing Documents is required for the variant *Goods Receipt with Production Order reference*.

Additional details for attribute - field mapping for billing and supplier invoice items can be found on [Billing Document Item | SAP Help Portal](#) and [Supplier Invoice Item | SAP Help Portal](#).

Loading Inventory

Open transaction code VPCOE/RDP and select service ‘Inventory’ (alternatively you can run the report /VPCOE/INVENTORY_TO_RDP):

Send Inventory Data to RDP

This is a TRIAL Version and only for DEMO purpose!

RDP Integration Accelerator V.5

Get Only Delta

Options

Selection Id: **RDP_INVEN-20240228**

Preview Data
 Send Data To Cloud
 Export to .XLSX File

Business Activity Metric Data

Company code	<input checked="" type="checkbox"/>	to	<input type="checkbox"/>	
Plant	<input checked="" type="checkbox"/>	to	<input type="checkbox"/>	
Products	<input type="checkbox"/>	to	<input type="checkbox"/>	
Year	<input type="checkbox"/>			
Month	<input type="checkbox"/>			

Additional Parameters

Storage Loc./Batch Stock
 Special Stock
 In Transit

Product Group: to 

Actual stock Inventory data will be extracted per material and plant for a specific key data (e.g. end of month).

Initial load approach for Inventory report:

- all the objects selected based on filters criteria will be sent to RDP
- data will be extracted **for specified document year/month** only (inventory on last day of the month)

Delta load approach for Inventory report:

- all the objects selected based on filters criteria will be sent to RDP
- All objects (Inventory stock) **for the specified month only** will be selected
- Year and Month are predefined with current year and month in delta load

Each month it will display the closing stock based on **material and company code/plant combination**. Every monthly extract contains all combinations of materials and company code/plant, even if the inventory quantity of that specific month is “0”. Negative stocks are not sent to RDP.

Also, please note that all stock type/categories are being considered.

Note: Company code, Year and Month are selection parameters that can be set on initial load and delta load also. All the other parameters are read-only on delta load.

Integration accelerator is using standard report MB5B to extract inventory data. If there are performance issues when selecting a large number of materials with storage location stocks (without batch or special stocks) it is recommended to implement sap note [2459529 - MB5B: Performance optimization during the selection of the storage location stock - SAP for Me](#)

and SAP note [1516684 - Enhancing MSEG with MKPF fields - performance optimization - SAP for Me](#)

Extensibility for RDP ERP Integration reports

The RDP Integration Accelerator provides extensibility through the BAdI **/VPCOE/ADJUST_DATA_RETRIEVAL**, enabling customer-specific adaptations for various integration scenarios.

1. Customer-Specific Adaptations

The BAdI **/VPCOE/ADJUST_DATA_RETRIEVAL** should be implemented to enhance RDP integration scenarios by:

- Adapting the **reading of source data** from other objects.
- Modifying the **mapping of data attributes** to interface attributes.
- Mapping custom values to ensure accurate data replication.

This BAdI is relevant if:

- Different source data are required for entity selection.
- Custom attributes need to be added during replication.

2. Applicable Reports

The following reports use the **/VPCOE/ADJUST_DATA_RETRIEVAL** BAdI:

Report Name	Description
/VPCOE/BATCH_TO_RDP	Send Batch data to RDP.
/VPCOE/CONFIGURATION_TO_RDP	Send Configuration data to RDP.
/VPCOE/CUSTOMER_TO_RDP	Send Customer data to RDP.
/VPCOE/DELIVERY_TO_RDP	Send Delivery data to RDP.
/VPCOE/INCOTERMS_TO_RDP	Send Incoterms data to RDP.
/VPCOE/INVENTORY_TO_RDP	Send Inventory data to RDP.
/VPCOE/MAT_DOC_TO_RDP	Send Material document data to RDP.
/VPCOE/ORG_DATA_TO_RDP	Send Organization data to RDP.
/VPCOE/PRODUCT_TO_RDP	Send Product data to RDP.
/VPCOE/SUPPLIER_TO_RDP	Send Supplier data to RDP.

Note: When any of these reports are executed, the BAdI implementation is triggered to apply customer-specific enhancements.

3. Interface and Methods in BAdI Implementation

The BAdI implements the interface **/VPCOE/IF_ADJ_DATA_RETRIEVAL** which contains the following methods:

Method Name	Description
ADJUST_DATA_RETRIEVAL	Adjust API data retrieval logic.
ADJUST_JSON	Modify the JSON payload based on customer-specific requirements.
ADJUST_MAPPING	Adapt or overwrite standard VPCOE mapping.
ADJUST_TEXT_MAPPING	Adjust mapping for text-based attributes.

GET_EXT_DATA	Retrieve extension data for Product and/or Customer scenarios.
SKIP_SELECTION	Skip standard data selection and apply custom selection logic.
ADJUST_BUILD_WHERE_FOR_VARIANT	Define custom WHERE conditions for selection logic.
DEFINE_MOVEMENT_TYPE	Implement custom movement type logic (for movement types 301, 303, 305).

4. Method Descriptions and Usage

4.1 SKIP_SELECTION

Allows skipping standard data selection logic.

If this method is implemented, the data retrieval logic must be fully customized using ADJUST_DATA_RETRIEVAL.

4.2 ADJUST_DATA_RETRIEVAL

Enables customer-specific data selection and adjustments before transmitting data.

4.3 ADJUST_JSON

Allows modifications to the JSON payload before sending data to RDP.

4.4 ADJUST_MAPPING

Supports custom mapping of attributes from the SAP backend to the RDP JSON structure.

4.5 GET_EXT_DATA

Retrieves additional extension data for Product and Customer replication.

4.6 ADJUST_TEXT_MAPPING

Provides customization for text-based attribute mappings.

4.7 ADJUST_BUILD_WHERE_FOR_VARIANT

Allows customization of WHERE conditions for selection logic, applicable for Material Document replication.

4.8 DEFINE_MOVEMENT_TYPE

- Specifies custom Z movement types corresponding to standard movement types **301, 303, and 305**.

Ensures correct data selection for custom movement types aligned with standard accelerator logic.

5. Handling Custom Movement Types in Material Documents

Background

- The Integration Accelerator does **not use movement types** as selection criteria by default.
- However, for movement types **301, 303, and 305**, additional logic is applied in the standard accelerator to ensure correct data selection.
- If **custom movement types (Z-types)** exist that correspond to these standard movement types, they must be defined explicitly in the BAdl using the method:

```
METHOD /VPCOE/IF_ADJ_DATA_RETRIEVAL~DEFINE_MOVEMENT_TYPE.
CASE iv_code.
  WHEN sc_variants-gr_st.
    mt_movement_type = VALUE #( ( sign = 'I' option = 'EQ' low = 'Z305' )
```

```

        ( sign = 'I' option = 'EQ' low = 'Z303' )
        ( sign = 'I' option = 'EQ' low = 'Z301' ) .
mt_stock_change_cat = VALUE #( ( sign = 'I' option = 'EQ' low = 'Z305' ) ) .

WHEN sc_variants-gi_st.
    mt_movement_type = VALUE #( ( sign = 'I' option = 'EQ' low = 'Z303' )
                                ( sign = 'I' option = 'EQ' low = 'Z301' ) ) .
ENDCASE.
ENDMETHOD.

```

Usage

- Use this method to map **custom movement types** to the standard types **301, 303 and/or 305** to ensure correct data retrieval and mapping.
- If no custom movement types are defined, this method does not require implementation.

6. Sample BAdI Implementation

A **sample BAdI implementation** is provided in class:

- **Class Name:** /VPCOE/CL_ADJUST_API_EXAMPLE

BAdI Implementation Guidelines:

- To **skip standard data selection**, set the parameter CV_SKIP = 'true' in **SKIP_SELECTION**. This ensures that the standard integration accelerator selection logic is bypassed, allowing full customization via /VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_DATA_RETRIEVAL
- If custom mapping of root nodes or subnodes is required, implement **ADJUST_MAPPING**.
- For direct JSON modifications, use **ADJUST_JSON**.
- In Material Document replication BAdI implementations, after defining custom selection logic in **ADJUST_DATA_RETRIEVAL**, it is mandatory to call **adjust_header** to ensure that ship-to and ship-from countries are correctly derived.
 - For further customization of country determination, implement **DEFINE_COUNTRY** accordingly.

7. Accessing BAdI Implementation

- To view or modify the sample implementation, use transaction **SE18** to explore the BAdI definition.
- For details of the parameters used in each method, open transaction **SE24**, enter the interface **/VPCOE/IF_ADJ_DATA_RETRIEVAL**, and execute it.

Enhancement Spot /VPCOE/ADJUST_API_TO_CLOUD Display

The screenshot shows the SAP Fiori interface for the Enhancement Spot /VPCOE/ADJUST_API_TO_CLOUD. The top navigation bar includes icons for back, forward, search, and other system functions. The main title is "Enhancement Spot /VPCOE/ADJUST_API_TO_CLOUD" with a status of "Active". Below the title, there are tabs: "Attributes", "Enhancem. Implementations", "Technical Details", and "Enh. Spot Element Definitions". The "Enh. Spot Element Definitions" tab is selected. On the left, a tree view shows "BAdI Definitions" expanded, with one item selected: "/VPCOE/ADJUST_DATA_RETRIEVAL". This item has two sub-options: "Interface" and "Implementations". The right panel contains several sections: "BAdI Definition" set to "/VPCOE/ADJUST_DATA_RETRIEVAL", "Description" set to "Adjust Data Retrieval", "Interface" set to "/VPCOE/IF_ADJ_DATA_RETRIEVAL", and "Usability" with a checked checkbox for "Multiple Use". There are also checkboxes for "Can only be implemented internally at SAP", "Limited filter use", and "AMDP BAdI". Under "Instance Creation Mode", the radio button "Creating instantiation" is selected. A checkbox for "Call fallback if no implementation is executed" is present, along with a "Fallback Class" input field. The "Implementation Example Classes" section lists one example class: "/VPCOE/CL_ADJUST_API_EXAMPLE" with a description of "Example BAdI Implementation.".

Sample Implementation

A sample BAdI implementation is provided in class /VPCOE/CL_ADJUST_API_EXAMPLE as a reference for custom development.

For more details, execute transaction SE18 to access the sample BAdI implementation.

Execute transaction SE18 to get more details regarding sample BAdI implementation.

BAdI Builder: Initial Screen for Definitions

The screenshot shows the BAdI Builder initial screen. At the top, there are icons for back, forward, search, and other system functions. Below the title, there are two radio button options: "Enhancement Spot" and "BAdI Name". The "BAdI Name" option is selected, and the input field next to it contains the value "/VPCOE/ADJUST_DATA_RETRIEVAL". At the bottom of the screen are three buttons: "Display", "Change", and "Create".

The screenshot shows the SAP Fiori Launchpad interface for the Enhancement Spot /VPCOE/ADJUST_API_TO_CLOUD. The main title bar says "Enhancement Spot /VPCOE/ADJUST_API_TO_CLOUD Display". Below it, there's a toolbar with various icons. The main content area has tabs: "Attributes", "Enhancem. Implementations", "Technical Details", and "Enh. Spot Element Definitions". The "Enh. Spot Element Definitions" tab is selected. On the left, there's a tree view under "BAdI Definitions" with one node expanded: "/VPCOE/ADJUST_DATA_RETRIEVAL / Adjust Data Retrieval". This node has two children: "Interface" and "Implementations". The right side of the screen contains several configuration sections. One section is titled "BAdI Definition" with the value "/VPCOE/ADJUST_DATA_RETRIEVAL". Another section is "Description" with the value "Adjust Data Retrieval". Under "Usability", there are checkboxes for "Multiple Use" (checked), "Can only be implemented internally at SAP" (unchecked), "Limited filter use" (unchecked), and "AMDP BAdI" (unchecked). A section for "Instance Creation Mode" has three radio buttons: "Creating instantiation" (selected), "Reusing Instantiation", and "Context-Specific Instantiation". There's also a checkbox for "Call fallback if no implementation is executed" which is unchecked. A "Fallback Class" field is present but empty. The final section is "Implementation Example Classes" with a table showing one example class: "/VPCOE/CL_ADJUST_API_EXAMPLE" with a description "Example BAdI Implementation."

Class Name: /VPCOE/CL_ADJUST_API_EXAMPLE

Badi Implementation Guidelines

- To **skip standard data selection**, set the parameter **CV_SKIP = 'true'** within **/VPCOE/IF_ADJ_DATA_RETRIEVAL~SKIP_SELECTION**. This ensures that the standard VPCOE selection logic is bypassed, allowing full customization via **/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_DATA_RETRIEVAL**.
- If **custom mapping** of root nodes or subnodes is required, use **/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_MAPPING**.
- For direct JSON modifications, implement **/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_JSON**.
- In Material Document replication BAdI implementations, after defining custom selection logic in **/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_DATA_RETRIEVAL**, it is **mandatory** to call **adjust_header**. This ensures that **ship-to** and **ship-from** countries are correctly derived for logistics processing.
 - If further customization of country determination is needed, implement **DEFINE_COUNTRY** accordingly.

To get further details of the parameters of each method, open transaction SE24, give the interface details **/VPCOE/IF_ADJ_DATA_RETRIEVAL** and execute it.

Class Builder: Display Interface /VPCOE/IF_ADJ_DATA_RETRIEVAL

Interface /VPCOE/IF_ADJ_DATA_RETRIEVAL Implemented / Active

Properties Interfaces Attributes Methods Events Types Aliases

Parameters Exceptions Filter

Method	Level	Description
ADJUST_DATA_RETRIEVAL	Instance Method	Adjust API data Retrieval
ADJUST_JSON	Instance Method	Adjust JSON data
ADJUST_MAPPING	Instance Method	Adjust Mapping
ADJUST_TEXT_MAPPING	Instance Method	Adjust Texts Mapping
GET_EXT_DATA	Instance Method	Get Extension Data
SKIP_SELECTION	Instance Method	Skip Selection Logic
ADJUST_BUILD_WHERE_FOR_VARIANT	Instance Method	Adjust where conditions for variant
DEFINE_MOVEMENT_TYPE	Instance Method	Definition of movement type

Select the method for which you want to see the parameter details and click on parameters, you will get all parameters details for that method.

Interface /VPCOE/IF_ADJ_DATA_RETRIEVAL Implemented / Active

Properties Interfaces Attributes Methods Events Types Aliases

Parameters of Method ADJUST_DATA_RETRIEVAL

Parameter	Type	P... O... Typing Method	Associated Type	Default Value	Description
IV_SRV_GRP	Importing	<input type="checkbox"/>	Type	/VPCOE/DE_SERVICE_GROUP	Service Group
IV_SRV_ID	Importing	<input type="checkbox"/>	Type	/VPCOE/DE_SERVICE_ID	Service ID
IV_API_TYPE	Importing	<input type="checkbox"/>	Type	/VPCOE/DE_API_TYPE	Api Type
IS_SEL_OPT	Importing	<input type="checkbox"/>	Type	DATA	
IV_LEVEL	Importing	<input type="checkbox"/>	Type	/VPCOE/LEVEL	Level in the JSON
IV_CODE	Importing	<input type="checkbox"/>	Type	CHAR20	Char 20
IO_LOG	Importing	<input type="checkbox"/>	Type Ref To	/VPCOE/CL_RDP_LOG	Log
IV_MODE	Importing	<input type="checkbox"/>	Type	/VPCOE/DE_MODE	Execution Mode
IV_SESSION_ID	Importing	<input type="checkbox"/>	Type	RAW16	RAW16
CT_DATA	Changing	<input type="checkbox"/>	Type	DATA	

Standard settings

For more information about the standard settings (filters, single or multiple uses), see the *Enh. Spot Element Definitions* tab in the *BAdl Builder* (transaction [SE18](#)).

See also

This BAdl uses the interface `/VPCOE/IF_ADJ_DATA_RETRIEVAL`.

For more information, display the interface in the [Class Builder](#) (Transaction [SE24](#)).

Additional BAdl documentation is available for the Badi:

Product and Customer extensions

Class `/VPCOE/CL_ADJUST_API_EXAMPLE` is sample implementation for your reference to implement the actual BAdI `/VPCOE/ADJUST_DATA_RETRIEVAL` for Product/Customer extension reports as per your requirement.

The method to be implemented for extension data is
`/VPCOE/IF_ADJ_DATA_RETRIEVAL~GET_EXT_DATA`

Method	Level	Visibility	Description
<code>/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_DATA_RETRIEVAL</code>	Instance Method	Public	Adjust API data Retrieval
<code>/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_JSON</code>	Instance Method	Public	Adjust JSON data
<code>/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_MAPPING</code>	Instance Method	Public	Adjust Mapping
<code>/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_TEXT_MAPPING</code>	Instance Method	Public	Adjust Texts Mapping
<code>/VPCOE/IF_ADJ_DATA_RETRIEVAL~GET_EXT_DATA</code>	Instance Method	Public	Get Extension Data
<code>/VPCOE/IF_ADJ_DATA_RETRIEVAL~SKIP_SELECTION</code>	Instance Method	Public	Skip Selection Logic

For more information on datatypes for extension attributes please have a look at 'RDP Fields Mapping v6.xls' file.

Also, additional details on all available attributes and its values for extension data are available here:

[Product Extension | SAP Help Portal](#)

Product Extension:

Column	API Element	SAP ECC/ S/4HANA Attribute	Description																										
Product Number	id	MATNR	ID of the product																										
Reference Quantity	referenceQuantity	DEC10	<p>Reference quantity for fee in base unit of measure.</p> <p>The reference quantity is used during fee calculation to provide enough decimals for an accurate calculation. The default value is 10,000 and the attribute refers to unit of measure of the product.</p>																										
Product Type harmonized	productTypeHarmonized	CHAR10	<p>Possible values:</p> <table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>FIN_GOOD</td><td>Finished good</td></tr> <tr> <td>RAWMATPACK</td><td>Raw material for packaging</td></tr> <tr> <td>PACKMAT</td><td>Packaging material</td></tr> <tr> <td>NOTAPPL</td><td>Not relevant</td></tr> <tr> <td>SEMIPACKM</td><td>Semifinished Packaging Material</td></tr> </tbody> </table>	Value	Description	FIN_GOOD	Finished good	RAWMATPACK	Raw material for packaging	PACKMAT	Packaging material	NOTAPPL	Not relevant	SEMIPACKM	Semifinished Packaging Material														
Value	Description																												
FIN_GOOD	Finished good																												
RAWMATPACK	Raw material for packaging																												
PACKMAT	Packaging material																												
NOTAPPL	Not relevant																												
SEMIPACKM	Semifinished Packaging Material																												
Product Content	productContent	CHAR10	<p>Possible values:</p> <table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>BEV_ALC</td><td>Alcoholic beverages</td></tr> <tr> <td>BEV_ALC_BE</td><td>Beer</td></tr> <tr> <td>BEV_OTHER</td><td>Other Beverages</td></tr> <tr> <td>BEV_WATER</td><td>Potable water</td></tr> <tr> <td>BEV_SOFT</td><td>Soft drinks</td></tr> <tr> <td>CHEMB2B</td><td>Chemicals in a B2B context (for example for industrial or agricultural use)</td></tr> <tr> <td>CLEANING</td><td>Cleaning products</td></tr> <tr> <td>CLOTHING</td><td>Clothing and apparel</td></tr> <tr> <td>COSM_CARE</td><td>Cosmetics and personal care</td></tr> <tr> <td>ELECTRONIC</td><td>Electronics</td></tr> <tr> <td>FOOD</td><td>Food and nutrition</td></tr> <tr> <td>FOOD_MED</td><td>Medical food</td></tr> </tbody> </table>	Value	Description	BEV_ALC	Alcoholic beverages	BEV_ALC_BE	Beer	BEV_OTHER	Other Beverages	BEV_WATER	Potable water	BEV_SOFT	Soft drinks	CHEMB2B	Chemicals in a B2B context (for example for industrial or agricultural use)	CLEANING	Cleaning products	CLOTHING	Clothing and apparel	COSM_CARE	Cosmetics and personal care	ELECTRONIC	Electronics	FOOD	Food and nutrition	FOOD_MED	Medical food
Value	Description																												
BEV_ALC	Alcoholic beverages																												
BEV_ALC_BE	Beer																												
BEV_OTHER	Other Beverages																												
BEV_WATER	Potable water																												
BEV_SOFT	Soft drinks																												
CHEMB2B	Chemicals in a B2B context (for example for industrial or agricultural use)																												
CLEANING	Cleaning products																												
CLOTHING	Clothing and apparel																												
COSM_CARE	Cosmetics and personal care																												
ELECTRONIC	Electronics																												
FOOD	Food and nutrition																												
FOOD_MED	Medical food																												

			<table border="1"> <tr><td>FOOD_DIR</td><td>Food for direct consumption</td></tr> <tr><td>MED_DEVICE or DEVICE_MED</td><td>Medical Electrical Device</td></tr> <tr><td>FURNITURE</td><td>Furniture, furnishings, and household items</td></tr> <tr><td>GAS</td><td>Gas, aerosol</td></tr> <tr><td>HOME</td><td>Home improvement and gardening</td></tr> <tr><td>NONFOOD</td><td>Other non-food</td></tr> <tr><td>PAINT</td><td>Paint</td></tr> <tr><td>PET</td><td>Pet food and animal supplies</td></tr> <tr><td>PHARMA</td><td>Pharmaceutical and medical goods (only for human medicine)</td></tr> <tr><td>SANI_WASTE</td><td>Intended to capture Hazardous Sanitary Waste</td></tr> <tr><td>TOBACCO</td><td>Tobacco</td></tr> <tr><td>TOYS</td><td>Toys</td></tr> <tr><td>VET_PHARMA</td><td>Veterinary pharmaceutical products (only for animal application)</td></tr> <tr><td>AGRI</td><td>Products designed for agricultural or livestock consumption</td></tr> </table>	FOOD_DIR	Food for direct consumption	MED_DEVICE or DEVICE_MED	Medical Electrical Device	FURNITURE	Furniture, furnishings, and household items	GAS	Gas, aerosol	HOME	Home improvement and gardening	NONFOOD	Other non-food	PAINT	Paint	PET	Pet food and animal supplies	PHARMA	Pharmaceutical and medical goods (only for human medicine)	SANI_WASTE	Intended to capture Hazardous Sanitary Waste	TOBACCO	Tobacco	TOYS	Toys	VET_PHARMA	Veterinary pharmaceutical products (only for animal application)	AGRI	Products designed for agricultural or livestock consumption
FOOD_DIR	Food for direct consumption																														
MED_DEVICE or DEVICE_MED	Medical Electrical Device																														
FURNITURE	Furniture, furnishings, and household items																														
GAS	Gas, aerosol																														
HOME	Home improvement and gardening																														
NONFOOD	Other non-food																														
PAINT	Paint																														
PET	Pet food and animal supplies																														
PHARMA	Pharmaceutical and medical goods (only for human medicine)																														
SANI_WASTE	Intended to capture Hazardous Sanitary Waste																														
TOBACCO	Tobacco																														
TOYS	Toys																														
VET_PHARMA	Veterinary pharmaceutical products (only for animal application)																														
AGRI	Products designed for agricultural or livestock consumption																														
Brand ID	brand	CHAR30	Brands consist of a name and an ID, they can be assigned to a product. You can create brands in the Configuration Cockpit application and assign brands to products in the Manage Products application.																												
Hazardous Product CLPEU	isHazardousClpEu	CHAR5	<p>Indicates if the product is classified as hazardous according to the CLP regulation.</p> <p>The CLP (Classification, Labeling, and Packaging) regulation is a European Union regulation that aligns the European Union legislation with the GHS (Globally Harmonized System of Classification and Labeling of Chemicals), which is managed by the United Nations.</p>																												
Product comes without packaging	hasNoPackaging	CHAR5	The product comes without packaging. This can apply to bulk products in a container without further packaging or the product is filled into returnable packaging brought by the end consumer, for example, a milk dispenser. These products do not need a packaging composition and can be ignored by most EPR																												

			reports. Packaging for products in this category is not distributed, collected, cleaned, or reused by the producer or retailer. Instead, they are fully under the control of the end-consumer and therefore would not require a packaging composition on the seller side. This model of packaging aims to avoid single-use packaging entirely.														
Product Countries	productCountries		<table border="1"> <thead> <tr> <th>API Element</th><th>ECC/S4H attribute</th></tr> </thead> <tbody> <tr> <td>country</td><td>LAND1</td></tr> <tr> <td>validFrom</td><td>DATS</td></tr> <tr> <td>tag</td><td>CHAR10</td></tr> <tr> <td>isNotEprRelevant</td><td>CHAR5</td></tr> <tr> <td>eprProductFamily</td><td>CHAR10</td></tr> <tr> <td>inhouseProductionPercent</td><td>DEC3</td></tr> </tbody> </table>	API Element	ECC/S4H attribute	country	LAND1	validFrom	DATS	tag	CHAR10	isNotEprRelevant	CHAR5	eprProductFamily	CHAR10	inhouseProductionPercent	DEC3
API Element	ECC/S4H attribute																
country	LAND1																
validFrom	DATS																
tag	CHAR10																
isNotEprRelevant	CHAR5																
eprProductFamily	CHAR10																
inhouseProductionPercent	DEC3																

Example of payload for Product extension:

```
{
  "source": "ECC",
  "elements": [
    {
      "id": "AFR-11-ART-001_____00011",
      "productTypeHarmonized": "PACKMAT",
      "productContent": "BEV_WATER",
      "isHazardousClpEu": true,
      "isDgProductEu": true,
      "hasNoPackaging": true,
      "brand": "1001",
      "referenceQuantity": "10000",
      "productCountries": [
        {
          "country": "DE",
          "validFrom": "2021-04-25",
          "tag": "IYSIUY765H",
          "isNotEprRelevant": true,
          "eprProductFamily": "id3",
          "inhouseProductionPercent": 13
        }
      ]
    }
  ]
}
```

Customer Extension:

Column	API Element	SAP ECC/S/4HANA Attribute	Description												
Customer Number	id	KUNNR	ID of the customer												
Tag	tag	Customer specific	Tag that can be assigned to customers. Customer tags can be created in the Configuration Cockpit application. Tags can be assigned to customers in the Manage Customers application.												
Role	role	Customer specific	A role that can be assigned to a customer. Possible values: <table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>END_CONSUM</td><td>End-consumer</td></tr> <tr> <td>RETAILER</td><td>Retailer</td></tr> <tr> <td>PROF_CUSTM</td><td>Professional customer</td></tr> <tr> <td>NEXT_PACKP</td><td>Next level packaging producer</td></tr> <tr> <td>HOSPITAL</td><td>Hospital</td></tr> </tbody> </table>	Value	Description	END_CONSUM	End-consumer	RETAILER	Retailer	PROF_CUSTM	Professional customer	NEXT_PACKP	Next level packaging producer	HOSPITAL	Hospital
Value	Description														
END_CONSUM	End-consumer														
RETAILER	Retailer														
PROF_CUSTM	Professional customer														
NEXT_PACKP	Next level packaging producer														
HOSPITAL	Hospital														

Example of payload for Customer extension:

```
{
  "source": "ECC",
  "elements": [
    {
      "id": "1000000001",
      "tag": "TAG",
      "role": "RETAILER"
    }
  ]
}
```

Please note, in case you don't have any value for some of the extension attributes, you need to implement the logic in BAdI method '/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_JSON' to replace empty values with NULL. Below is small example of the code you may introduce for this matter (a similar example it's also on the sample BAdI implementation we are providing):

In BAdI method '/VPCOE/IF_ADJ_DATA_RETRIEVAL~ADJUST_JSON', introduce code below:

```
IF      iv_api_type = /vpcoe/cl_customizing_helper=>sc_api_type-rdp
AND    iv_srv_grp  = /vpcoe/cl_customizing_helper=>sc_grp_id-product
AND    iv_srv_id   = /vpcoe/cl_customizing_helper=>sc_service_id-PRODUCT_EXT.

LOOP AT ct_json ASSIGNING <ls_json>.
  REPLACE ALL OCCURRENCES OF ':""' IN <ls_json>-elements WITH ': null'.
ENDLOOP.
ENDIF.
```

For customer extension *role* attribute, the sample badi impelmentation contains the logic based on RDP recomendation on class and characteristics:

Class	ZCCL_RDP_CUST_EXT
Description	RDP specific Customer Attributes
Class Type	011

Characteristics	Description	Data Type	Length	Decimal	UoM / Values
ZRDP_CUST_EXT_ROLE	Customer Role	CHAR	10	0	Code List
ZRDP_PROD_EXT_TAG	Customer Tag	CHAR	10	0	Customer defined code list

Email notification adaptation

Whenever replication is failing for an entity, if customer maintained [email addresses configuration settings](#), an email will be sent to all maintained email addresses.

BAdI `/VPCOE/NOTIFICATION_PROCESSING` is available for adapting the email sender / email body content or for overwriting error notification processing.

For more details and example implementation you can have a look at sample BAdI implementation `/VPCOE/CL_NOTIFICATION_EXAMPLE`.

Excel documents processing adaptation

BAdI `/VPCOE/XLS_PROCESSING` is available for excel documents processing adjustments.

/VPCOE/XLS_PROCESSING – technical documentation

The Business Add-In (BAdI) `/VPCOE/XLS_PROCESSING` that is assigned to Enhancement Spot `/VPCOE/ADJUST_API_TO_CLOUD`, this allows to override the logic for saving the Excel file. With this BADI, you can add your own logic for saving files both in the background and in real time, or completely change it. It also provides the option to change the save path of the file, if necessary.

This BAdI is used in the "`/VPCOE/CL_XLS_HANDLER`" class in the "`SAVE_XLS_FILE`" method.

Excel file download is available in the following reports:

- `/vpcoe/batch_to_rdp` Send Batch data to the RDP cloud
- `/vpcoe/configuration_to_rdp` Send Configuration data to the RDP cloud
- `/vpcoe/customer_to_rdp` Send Customer data to the RDP cloud
- `/vpcoe/delivery_to_rdp` Send Delivery data to the RDP cloud
- `/vpcoe/incoterms_to_rdp` Send Incoterms data to the RDP cloud
- `/vpcoe/inventory_to_rdp` Send Inventory data to the RDP cloud
- `/vpcoe/mat_doc_to_rdp` Send Material data to the RDP cloud
- `/vpcoe/org_data_to_rdp` Send Organization data to the RDP cloud
- `/vpcoe/product_to_rdp` Send Product data to the RDP cloud
- `/vpcoe/supplier_to_rdp` Send Supplier data to the RDP cloud

For more information about the standard settings for this BAdI see the Spot Element Definitions tab in the BAdI Builder (transaction SE18).

BAdI Methods

The BAdI definition is active in the Integration accelerator package. It contains the following method from interface `/VPCOE/IF_XLS_PROCESSING`:

· ADJUST_FILE_SAVING

As described above, this method allows to rewrite the logic for saving the Excel file or change the path that was specified on the selection screen.

The method uses the following parameters:

- IV_BACKGROUND : An import parameter that is used to determine in which mode the file will be saved
- IO_LOG : An import parameter for writing messages to the LOG
- EV_SKIP_STANDARD_PROCESS : The export parameter. If the value of the parameter is "true", then we skip the standard implementation of the "SAVE_XLS_FILE" method.
- CV_FILE_PATH : Changing parameter. This parameter is passed the path where the file is saved, as specified in the selection screen. Since the type of this parameter is "Change", you can change it.
- CT_XLS_DATA : Change parameter. This is a table that stores all the information of an Excel file in the Binary type

BAdI Implementations

The BAdI implementation must be defined by the user and is not provided by SAP.

Sample implementation

As example see class /VPCOE/CL_XLS_PROCESSING_EXMPL.

Resending failed payloads

If during data replication there are failed payloads from various reasons, all these are recorded, and you will have the possibility to resend them later. Please note if there are missing data into your records you cannot just resend the payloads, you will have to correct the data and use specific data reports to reselect and send the data again.

You can run the report /VPCOE/REPROCESS_PAYLOAD using SE38 transaction code:

ReProcess Failed Payloads

This is a TRIAL Version and only for DEMO purpose!

API Type	to	repeat
Service Group	to	repeat
Service Id	to	repeat
Session Id	0000000000000000... to 0000000000000000...	repeat
Session Item	to	repeat
Created Date	to	repeat

Processing Mode

Automatic
 Manual

You can choose from two options for resending the failed payloads: manual or automatic mode.

By choosing automatic mode, all failed payloads for specific filter criteria used are resent automatically. This option can be chosen when executing the report in scheduled jobs.

In manual mode, you have to manually select specific request(s) from the list and click on 'Repeat request' button to resend a request.

If the requests are sent successfully, they will be deleted from the list.

ReProcess Failed Payloads

Also, in case there is a failed payload that needs to be analyzed further, there is the option to display and copy/save the failed payload for further processing (e.g using Postman tool).

RDP Packaging Data Integration reports

Packaging data, such as packaging compositions, packaging elements, and basic fractions that are uploaded and evaluated in SAP Responsible Design and Production can be stored in SAP ECC or SAP S/4HANA in different locations, for example, in SAP Product Lifecycle Management (PLM), SAP Recipe Management (RM) or Bill of Material (BOM). Therefore, this integration feature offers only basic functionality to extract the packaging data out of these storage locations. Customer specific program extensions and a BAdl implementation, where these program extensions are referenced, are required. If no customer specific program extensions are implemented, the upload reports will not upload any packaging data to SAP Responsible Design and Production.

Reports included:

- Report for Uploading Packaging Elements from PLM:
/VPCOE/R_UPH_PCKG_DATA_LOAD
 - Report for uploading Packaging Compositions from PLM:
/VPCOE/R_UPH_PCKG_CMP_PLM_LOAD
 - Report for Uploading Packaging Compositions from Recipe Management:
/VPCOE/R_UPH_PCKG_CMP_RCP_LOAD
 - Report for Uploading Packaging Composition from BoM
/VPCOE/R_UPH_PCKG_CMP_BOM_LOAD
 - Report for Uploading Packaging elements from Material Classification
/VPCOE/R_UPH_PCKG_MCL_LOAD

- The BAdI provided for Upload Entity processor implementation:
/VPCOE/BADI_UPH_CUSTOM (use enh. spot /VPCOE/ES_UPH_CUSTOM)

To upload or push data out of the ECC/S4HANA to the SAP RDP solution, the following software parts are necessary:

1. Report(s) to start the upload of data (specification database data or other) to RDP endpoint
2. Implementation of Upload Entity
 - o Gathering of current relevant data (full or delta data) of the Upload Entity (if necessary, provide and use BADIs for customer extensibility)
 - o Packaging the relevant data
 - o Map the gathered data to internal data structures
 - o Transmit the data stored in internal data structures to the corresponding RDP-Push API
 - o For Initial/delta upload, store the information about uploaded data

An **Upload Entity** represents an RDP business object (like Packaging Element) with all its defined child nodes (like Packaging Fractions, Product assignments). The structure of an Upload Entity must fit to the provided RDP OData Entity structure for writing data.

Here you can find more details on extensibility and how to implement entity processor class.

Creating recommended classes and characteristics

To facilitate effective packaging data management within SAP RDP, a specialized report has been included to generate recommended classes and characteristics. The report is creating the classes and characteristics for:

- o Packaging Elements in Material Classification
- o Packaging Compositions in BOM, classifying Pckg. Elmnts. into Main/Subordinate
- o Packaging Compositions and Main/Subordinate Packaging Elements in PLM SpecDB
- o Packaging fee data classification
 - Class: ZRDP_ES_PLASTICTAX
 - Characteristics:
 - ZRDP_ES_PLASTIC
 - ZRDP_ES_NONRECYCLED_PLASTIC
 - ZRDP_ES_EXEMPTION
 - ZRDP_ES_DECREFNO
 - ZRDP_ES_CERTREFNO
 - ZRDP_ES_PRODORDER_CHECK
 - ZRDP_ES_PACKMAT_TAX
- o Customer extension data
 - Class: ZRDP_RDP_CUST_EXT
 - Characteristics:
 - ZRDP_CUST_EXT_ROLE
 - ZRDP_PROD_EXT_TAG

The report name is /VPCOE/RDP_CLAS_DATA_CREATOR.

The transaction code /VPCOE/DATA_CREATOR belongs to this report execution.

There are two parameter sections: Set Prefix, and Set Operation Scope.

In the Set Prefix section, set prefixes for classifications and characteristics. Prefixes must not be empty and can't be longer than four characters.

In the Set Operation Scope section, choose between Material Classification and Specification Workbench. The prefixes will be reset automatically to default values that match the default implementation of class /VPCOE/CL_PCKELEM_MCL_EXMPL.

After the execution the program will show the results of created and not created classifications and characteristics.

To modify classifications and characteristics that the program creates, you need to create own Z-objects and copy the code of the report and class /VPCOE/CL_CL_CHAR_DATA_CREATE, which the report calls upon execution.

In your own Z-class you have the following options:

- Changing the classifications and characteristics for packaging element **fractions**: Go to the `setup_pckelem_frac_dataset()` method and modify `lt_pckfrac_classi_params` and `lt_pckfrac_charac_params`.
- Changing the classifications and characteristics for packaging element **attributes**: Go to the `setup_pckelem_attr_dataset()` method and modify `lt_pckattr_classi_params` and `lt_pckattr_charac_params`.

Please note that your modifications must match modifications in example class /VPCOE/CL_PCKELEM_MCL_EXMPL.

Upload Packaging Elements data from PLM

The upload of packaging element data from PLM (Product Lifecycle Management) requires that the according data exists in the Environment, Health, and Safety specification database. In addition, an Upload Entity Processor implementation for the respective Upload Entity PE_PLM exists in the system and is returned by the BAld implementation of /VPCOE/BADI_UPH_CUSTOM.

The Upload Entity: Packaging data (PLM) retrieves the packaging related data out of the Specification Management.

The report name is /VPCOE/R_UPH_PCKG_DATA_LOAD.

The transaction code /VPCOE/UPH_PE_PLM belongs to this report execution.

The target internal data structures for the retrieved data are implementations of interface /VPCOE/IF_UPH_ENTITY_DATA.

Processor classes are implementations of interface /VPCOE/IF_UPH_ENTITY_PROC and in addition for the PLM case of interface /VPCOE/IF_UPH_ENTITY_PLM_PROC. The base class /VPCOE/CL_UPH_PROC_BASE_PLM provides a basic implementation of these processor interfaces and must be extended for specific adoptions.

Class overview

Interface/Class	Description
/VPCOE/IF_UPH_ENTITY_PROC	<i>Interface for Upload Entity Processors</i>
/VPCOE/IF_UPH_ENTITY_PLM_PROC	<i>Interface for Entity Processors based on PLM data</i>
/VPCOE/CL_UPH_PROC_BASE	<i>Abstract base processor implementation.</i>

	<i>Concrete data extraction implementations must inherit from this base class.</i>
/VPCOE/CL_UPH_PROC_BASE_PLM	<i>Base processor implementation for Upload Entity Packaging PLM. Concrete PLM data extraction implementations must inherit from this base class.</i>
/VPCOE/CL_UPH_WRAP_SPEC	<i>Wrapper Class for a Specification</i>
/VPCOE/CL_UPH_WRAP_VA	<i>Wrapper Class for Value Assignment data (with Characteristics) of a Specification</i>
/VPCOE/IF_UPH_ENTITY_DATA	<i>Interface for Upload Entity data</i>
/VPCOE/CL_UPH_ENT_PCKG_ELEMENT	<i>Entity data for Packaging Element</i>
/VPCOE/CL_UPH_ENT_PCKG_FRCTN	<i>Entity data for Basic Material Fraction</i>
/VPCOE/CL_UPH_ENT_PCKG_PRODUCT	<i>Entity data for Packaging Element to Product</i>
/VPCOE/IF_UPH_ENTITY_HU_PROC	<i>Interface for Upload Handling Units</i>
/VPCOE/CL_UPH_ENT_HANDL_UNIT	<i>Entity data for Packaging Composition Items (Handling Units)</i>

Extensibility

Implementing the Upload Entity Processor

An Upload Entity Processor implementation for the Upload Entity PE_PLM should inherit from base class /VPCOE/CL_UPH_PROC_BASE_PLM. This base class is already implementing the methods of the interface /VPCOE/IF_UPH_ENTITY_PROC and only the mapping of the specification data must be provided by implementing the method /VPCOE/IF_UPH_ENTITY_PLM_PROC~map_spec_data.

See the **Technical Documentation** of BAdI /VPCOE/BADI_UPH_CUSTOM for an example implementation of such an Upload Entity Processor.

The implementation of custom entity processor can be realized by the following steps:

1. Create a processor implementation by implementing the interface /VPCOE/IF_UPH_ENTITY_PROC. If the processor is used to read out of the specification management, derive the implementation from the basis class /VPCOE/CL_UPH_PROC_BASE_PLM.
 - implement method /VPCOE/IF_UPH_ENTITY_PROC~prepare_process to determine which **specifications are relevant**. Based on data input on the selection screen, the relevant specifications for the data upload are determined. When running in delta processing mode, here the relevant specifications must be filtered to only the changed one (since the last processing).
 - implement all **relevant Value Assignment Types** which contain Packaging Element or Fraction data in method /VPCOE/IF_UPH_ENTITY_PLM_PROC~get_relevant_vat
 - All **mappings** between specification data and the target Entity Data (Packaging Element, Packaging Fraction, ...) must be implemented in the method /VPCOE/IF_UPH_ENTITY_PLM_PROC~map_spec_data
2. Exchange the default BADI implementation for /VPCOE/BADI_UPH_CUSTOM and deliver the new implementation.

Last change information

There is no CDS views (internal API) that used in the S/4 HANA implementation to get some of the required data. That's why separate class has been implemented for ECC solution - /VPCOE/CL_PLM_DB_ACCESS.

Upload Packaging Compositions from PLM

The upload of packaging composition data from PLM requires that the accompanying data exists in the Environment, Health, and Safety specification database. In addition, an Upload Entity Processor implementation for the respective Upload Entity PC_PLM exists in the system and is returned by the BAdI implementation of VPCOE/BADI_UPH_CUSTOM. The Report name is /VPCOE/R_UPH_PCKG_CMP_PLM_LOAD.

The transaction code /VPCOE/UPH_PC_PLM belongs to this report execution.

Implementing the Upload Entity Processor

An Upload Entity Processor implementation for the Upload Entity PC_PLM should inherit from base class /VPCOE/CL_UPH_PROC_BASE_PLM. This base class is already implementing the methods of the interface /VPCOE/IF_UPH_ENTITY_PROC and only the mapping of the specification data must be provided by implementing the method /VPCOE/IF_UPH_ENTITY_PLM_PROC~MAP_SPEC_DATA from interface VPCOE/IF_UPH_ENTITY_PLM_PROC.

See the **Technical Documentation** of BAdI VPCOE/BADI_UPH_CUSTOM for an example implementation of such an entity processor.

Class overview

The objects created for packaging elements could be reused for the extraction of packaging compositions. The list below supports the statement:

Report	/VPCOE/R_UPH_PCKG_CMP_PLM_LOAD	Extract Packaging Composition from Spec DB
Factory Class	SAME - Add new transfer mapper for the composition	
BADI	No Change Needed - Call right upload entity from the entity processor	
BASE	No Change Needed	
BASE PLM	No Change Needed	
WRAP_SPEC	No Change Needed	
WRAP_VA	No Change Needed	
Example Class Implementation	/VPCOE/ZCL_UPH_CUSTOM_IMPL	Call the right example implementation;
	/VPCOE/CL_UPH_PROC_PLM_EXMPL1	Custom/Example Implementation for Extraction of Packaging Composition from Spec DB
Logger class	No change	

Access the Specifications data

There is no CDS views (internal API) that used in the S/4 HANA implementation to get some of the required data. So, separate class is implemented for ECC solution -
 /VPCOE/CL_PLM_DB_ACCESS.

To get Specification data the following methods are used:

- GET_SPECIFICATIONFORKEYDATE
- GET_SPECASSIGNEDMATLFORKEYDATE

Upload Packaging Compositions from Recipe Management

Retrieves the packaging composition related data out of the Recipe Development.

The Report is /VPCOE/R_UPH_PCKG_CMP_RCP_LOAD.

The transaction code /VPCOE/UPH_PC_RCP belongs to this report execution.

Class overview

The objects created for packaging elements and packaging components from Spec DB could be reused for the extraction of packaging compositions from Recipe development. The list below supports the statement:

Report	/VPCOE/R_UPH_PCKG_CMP_RCP_LOAD	Extract Packaging Composition from Recipe Development; Report screen fields are mentioned in the table below
Factory Class	No Change Needed	
BADI	No Change Needed	
BASE	No Change Needed	
BASE PLM	/VPCOE/CL_UPH_PROC_BASE_RCP	Base Class to retrieve packaging composition data from Recipe development
WRAP_SPEC	/VPCOE/CL_UPH_WRAP_CMP_RCP	Wrapper class to retrieve packaging composition data from recipe
WRAP_VA		
Example class Implementation	/VPCOE/ZCL_UPH_CUSTOM_IMPL	Call right example implementation
	/VPCOE/CL_UPH_CMP_RCP_EXAMPLE	Custom/Example Implementation for Extraction of Packaging Composition from Recipe development
Class for Comp_header	/VPCOE/CL_UPH_ENT_PCKG_CMP_HDR	All of these Packaging Composition elements from Spec DB would be reused
Class for Comp_Item	/VPCOE/CL_UPH_ENT_PCKG_CMP_ITM	

Transfer Mapper class	/VPCOE/CL_UPH_ENT_PCKG_CMP_ITM	
Logger class	No change needed	Adapt for messages specific to recipe development

Access the Recipe Management data

There is no CDS views (internal API) that used in the S/4 HANA implementation to get some of the required data. So, separate class is implemented for ECC solution -

/VPCOE/CL_PLM_DB_ACCESS.

To get Specification data the following methods are used:

- GET_RECIPESTATUSTEXT
- GET_SINGLE_RECIPETYPE
- GET_RECIPEFORKEYDATE
- GET_RECIEFFORMULAITEM

Uploading Packaging Compositions from Bills of Material

The upload of packaging composition data from Bills of Material (BOMs) requires that the according data exist in the bill of material. In addition, an Upload Entity Processor implementation for the respective Upload Entity PC_BOM exists in the system and is returned by the BAdI implementation of /VPCOE/BADI_UPH_CUSTOM.

The Report is /VPCOE/R_UPH_PCKG_CMP_BOM_LOAD.

The transaction code /VPCOE/UPH_PC_BOM belongs to this report execution.

Implement the Upload Entity Processor

An Upload Entity Processor implementation for the Upload Entity PC_BOM should inherit from base class /VPCOE/CL_UPH_PROC_BASE_BOM. This base class is already implementing the methods of the interface /VPCOE/IF_UPH_ENTITY_PROC and only the mapping of the BOM must be provided by implementing the method /VPCOE/IF_UPH_ENTITY_BOM_PROC~MAP_BOM_DATA.

See the **Technical Documentation** of BAdI /VPCOE/BADI_UPH_CUSTOM for an example implementation of such an entity processor.

Report parameters:

- **Initial Load / Delta Load** (Radio Buttons):

If the Initial Load radio button is selected, then the report selects the BOMs based on the selection screen parameters.

If the Delta Load radio-button is selected, then the selection screen parameters of the previous successful initial load are used and therefore all selection screen parameters except the Test Mode are hidden. In addition to that, during delta load, only those BOMs that have been updated since the last successful initial or delta load will be extracted and transferred.

- **Test Mode** (Check Box):

If the Test Mode check box is selected, then the entire processing in terms of extraction and mapping of BOM data remains same only the transferring of the data to the SAP Responsible Design and Production API is skipped. Test Mode is supported in initial load or delta load.

- **Source ID (Input Parameter):**

Source ID is a mandatory input parameter which is transferred to SAP Responsible Design and Production. It identifies from which source system the data is coming.

- **RFC Destination (Input Parameter):**

This parameter is mandatory and contains the customer implemented target destination for the SAP Responsible Design and Production API (defined in transaction sm59). The default value VPCOE_RDP_PLM can be changed here.

- **Use Filter Criteria:**

The following input parameters are enabled for data filtering.

- Material (include ranges, patterns, multi selection): Filter for BOMs that are assigned to this material
- Material type (include ranges, patterns, multi selection): Filter for BOMs that are assigned to materials of this material type
- Plant (include ranges, patterns, multi selection): Filter for BOMs that are assigned to this plant
- BOM Usage (include ranges, patterns, multi selection): Filter for BOMs that are defined for this usage area
- Alternative BOM (include ranges, patterns, multi selection): Filter for BOMs with this alternative number
- BOM Status (single selection): Filter for BOMs with this status
- BOM Valid From Date: This field is used to filter the BOM validity. All BOMs that are valid before or on this date will be selected for processing. By default, this field is hidden and has the value '31.12.9999' which means that every version will be selected for processing.
- Change Date of BOM (include ranges, multi selection): Filter for BOMs with this (overall) change date

Use the DISPLAY_VALIDITY function code to see this field. You can change the value but consider that changing the BOM Valid From Date can lead to missing links between packaging elements and packaging compositions when not all versions are uploaded.

Additional Settings: The following additional settings configure the search.

- Explosion Level (Input parameter): A mandatory parameter that defines the maximum number of levels that are used to explode the BOM. This is relevant for nested BOMs only.

BOM Valid From Date:

Following scenarios shall be supported:

1. Valid From date is provided: All BOM that are valid for this key date shall be replicated. Apply this logic for both, initial load and delta load.
2. Valid From date is visible and is left empty: Empty from-date means that from-date = current date. All BOM that are valid for this rolling key date shall be replicated. Apply this logic for both, initial load and delta load.
3. Valid From date is hidden: the default value for this field is '31.12.9999' which means that every version will be selected for processing.

Change Date of entire BOM

Following scenarios are supported:

1. Provide only from-date, only BOM header and components changed on that specific date are extracted (whole BOM is extracted)
2. Provide from-date and to-date, all BOM header and components changed between (>= from-date and <= end-date) are extracted

Default Values for Selection Parameters:

- The BOM Valid From Date is always defaulted to "31.12.9999"
- Radio button for Initial load is defaulted.

Mandatory Selection Parameters for Filtering Packaging Compositions

If extracting by Filter Criteria:

- The field "Maximum explosion level" shall be mandatory with default value 10.

The upload of packaging composition data from Bills of Material (BOMs) requires that the according data exist in the bill of material. In addition, an Upload Entity Processor implementation for the respective Upload Entity PC_BOM exists in the system and is returned by the BAdI implementation of /VPCOE/BADI_UPH_CUSTOM.

Use report /VPCOE/R_UPH_PCKG_CMP_BOM_LOAD to extract packaging composition data from BOMs and to push it to the SAP Responsible Design and Production.

An Upload Entity Processor implementation for the Upload Entity PC_BOM should inherit from base class /VPCOE/CL_SURDP_UPH_PROC_BASE_BOM. This base class is already implementing the methods of the interface /VPCOE/IF_UPH_ENTITY_PROC and only the mapping of the BOM must be provided by implementing the method /VPCOE/IF_UPH_ENTITY_BOM_PROC~MAP_BOM_DATA.

Class Overview

Report	/VPCOE/R_UPH_PCKG_CMP_BOM_LOAD	Extract Packaging Composition from BoM
---------------	--------------------------------	--

Factory Class	SAME - Add new transfer mapper for the composition	
BADI	No Change Needed - Call right upload entity from the entity processor	
WRAP_SPEC	No Change Needed	
WRAP_VA	No Change Needed	
Implementing Class	/VPCOE/ZCL_UPH_CUSTOM_IMPL	Call the right example implementation;
Example Implementation	/VPCOE/CL_UPH_CMP_BOM_EXAMPLE	Custom/Example Implementation for Extraction of Packaging Composition from BoM
Logger class	No change	

Sample BAdI Implementations for V1 and V2 RDP Packaging data APIs

Sample BAdI implementations are provided for both **V1** and **V2** of the Packaging Data API. These implementations offer a comprehensive framework for processing and transferring BOM (Bill of Materials) data as part of the RDP Integration Accelerator.

Key Features Included in the Sample BAdI Implementation

1. **BOM Extraction Based on Filters**
 - Extracts BOMs based on the filter criteria defined in the report.
2. **Material Type Filtering**
 - Filters BOM positions by materials of **material type VERP** (packaging materials).
3. **Classification and Assembly BOM Checks**
 - For each BOM position:
 - Checks if a **material classification** for packaging data exists.
 - Validates the **valid-from date** of the BOM position.
 - Checks if the BOM position is an **assembly BOM** with an existing batch classification as the main element.
4. **Transfer as Packaging Composition Item**
 - If a **packaging data classification** exists, the BOM position is transferred as a **packaging composition item**.
5. **Combination of VERP Positions in Assembly BOMs**
 - Searches through all assembly BOMs for positions of **type VERP**.
 - Combines the **VERP positions** into a single **packaging composition** for the top-level BOM header (Product).
6. **Handling Assembly BOMs with Batch Classification**
 - In case of an assembly BOM with an **existing batch classification** as the main element:
 - Positions within the assembly are considered as **subordinate elements**.
 - The presence of **packaging data material classification** and **batch classification** for subordinate elements is required.
7. **Exclusion of BOM Positions without Classification**
 - BOM positions are **not transferred** if no classification exists.

Additional Considerations:

- Both **V1** and **V2** implementations allow for further customization if required by the business.
- The sample BAdI implementations can be adapted to meet specific requirements by modifying the standard logic provided in the default classes.

The upload of packaging element data from Material Classification requires that the according data exists in the 'Material'. In addition, an Upload Entity Processor implementation for the respective Upload Entity PE_MCL exists in the system and is returned by the BAdl implementation of /VPCOE/BADI_UPH_CUSTOM.

The Report name is /VPCOE/R_UPH_PCKG_MCL_LOAD.

The transaction code /VPCOE/UPH_PCKG_MCL belongs to this report execution.

Implementing the Upload Entity Processor

An Upload Entity Processor implementation for the Upload Entity PE_MCL should inherit from base class /VPCOE/CL_UPH_PROC_BASE_MCL. This base class is already implementing the methods of the interface /VPCOE/IF_UPH_ENTITY_PROC.

The mapping of the material classification must be provided by implementing the method /VPCOE/IF_UPH_ENTITY_MCL_PROC~MAP_MAT_CLAS_DATA to determine which classification is relevant for the business case.

Method /VPCOE/IF_UPH_ENTITY_MCL_PROC~GET_RELEVANT_MAT_CLAS must be also implemented.

For an example implementation of this entity processor, see class /VPCOE/CL_UPH_PCKELEM_MCL_EXMPL and the contained ABAP documentation.

Report parameters

- Initial Load / Delta Load (Radio buttons):

If the Initial Load radio button is selected, then the report selects the material classifications based on the selection screen parameters.

When the Delta Load radio-button is selected then the previous successful initial load of the selection screen parameters is used and therefore all selection screen parameters except the Test Mode are hidden. In addition to that, during delta load, only those specifications that have been updated since the last successful initial or delta load will be extracted and transferred.

- Test Mode (Check box):

If the Test Mode check box is selected, then the entire processing in terms of extraction and mapping of material classifications remains same, only the transferring of the data to the SAP Responsible Design and Production API is skipped. Test Mode is supported in initial load and delta load.

- Source ID (Input parameter):

Source ID is a mandatory input parameter that is transferred to SAP Responsible Design and Production. It identifies from which source system the data is coming.

- RFC Destination (Input parameter):

This parameter is mandatory and contains the customer implemented target destination for the SAP Responsible Design and Production API (defined in transaction SM59). The default value RDP_REPL_API can be changed here.

- Use Filter Criteria: The following input parameters are input enabled for data filtering:
 - Material (include ranges, patterns, multi selection): Filter for material classification that are assigned to this material
 - Material type (include ranges, patterns, multi selection): Filter for material classifications that are assigned to materials of this material type
 - Change Date of Whole Spec (include ranges, multi selection): Filter for materials that were last changes on these dates

Handling of Multiple Versions

Material Classification supports multiple versions with date and time related validity via Change Numbers. However, the report always extracts a packaging element out of the Material Classification as it is valid on the day of execution of the replication.

Change Date of Material

Following scenarios are supported:

1. If only from-date is provided, only materials changed **on** that specific date are extracted. Also, the change of a characteristic value from the material classification is considered as change.
2. If from-date and to-date are provided, all materials changed between (\geq from-date and \leq end-date) are extracted. Also, the change of a characteristic value from the material classification is considered as change.

Default Values for target RFC destination Parameters

- The parameter should have the customer implemented target destination as default. If the customer does not have custom implementation for target RFC destination, then VPCOE_RDP_PLM_PKG_ELMNT as default value but can be adjusted by the user. This gives customers more flexibility in choosing a destination name and avoids potential name conflicts.

Class Overview

Report	/VPCOE/R_UPH_PCKG_MCL_LOAD	Extract Packaging elements from Mat. Classification
Factory Class	SAME - Add new transfer mapper for the composition	
BADI	Call right upload entity from the entity processor	
WRAP_SPEC	No Change Needed	

WRAP_VA	No Change Needed	
Implementing Class	/VPCOE/ZCL_UPH_CUSTOM_IMPL	Call the right example implementation;
Example Implementation	/VPCOE/CL_UPH_PROC_MCL_EXAMPLE	Custom/Example Implementation for Extraction of Packaging Element from Material Classification
Logger class	No change	

Uploading Packaging Composition Items from Handling Units

This report is used when it is necessary to **add additional packaging elements** to an existing packaging composition in SAP Responsible Design and Production (RDP).

Use Case

- Products often require **tertiary packaging materials** (e.g., pallets, stretch foil) to protect and secure finished products during transport.
- Such packaging materials are typically bundled as **handling units (HUs)**. For example:
 - A **pallet** with a dozen cardboard boxes.
 - Each box containing individually packed finished goods.
 - The pallet and boxes are wrapped with a few meters of stretch foil.

How the Report Works

- The report looks up the **effectively used packaging material** from handling units in **deliveries** within a specified timeframe.
- It adds the corresponding **packaging elements** to the existing **packaging compositions** in RDP, each element with its **proportionate quantity**.
- This enables RDP to identify how much of each packaging material is used per finished product unit.

Handling Variations in Packaging Material Quantities

- It's possible that deliveries with the same packaging composition may have slightly different packaging material quantities.
- To account for such variations, the report **calculates the average** over the defined period to provide a more accurate estimate.

Note

- Currently, the report **does not support** transferring data to **RDP Packaging API V2** by default.
- However, this functionality can be implemented through a **custom BAdI implementation** if needed.

Extensibility and Custom Implementation

Due to the variety of handling unit configurations, the report offers extensibility options for:

- **Extracting handling units** using custom logic.
- **Updating packaging compositions** with additional items.

Upload Entity Processor for PCI_HU

- An **Upload Entity Processor** implementation for the respective entity PCI_HU must be present in the system.
- This implementation is returned by the BAdI **/VPCOE/BADI_UPH_CUSTOM**.

Report for Uploading Handling Unit Data

- **Report Name:** /VPCOE/R_UPH_HU_DATA_LOAD
- **Purpose:** Replicate packaging composition items derived from handling units to RDP.

Prerequisites and Limitations

Prerequisites

- To use the report for uploading packaging composition items:
 - Create a **Replication Configuration** in RDP through the **Configuration Cockpit** for the object type **Packaging Composition – Additional Items**.

Limitations and Considerations

When implementing the BAdI for uploading additional packaging composition items, consider the following limitations:

- **Separate Source System Requirement:**
 - Replication configurations for **Packaging Composition** and **Packaging Composition – Additional Items** cannot be created in the same source system.
 - A replication configuration for **Packaging Composition – Additional Items** must refer to a system **not yet used**.
- **Handling Data from the Same System:**
 - If data for **Packaging Composition** is pushed from the same system:
 - Create a replication configuration with a **different Source System ID**.
 - Use this alternative name only when calling the API for **packaging composition items**.

Implement the Upload Entity Processor

An Upload Entity Processor implementation for the Upload Entity PC_HU should inherit from base class /VPCOE/CL_UPH_PROC_BASE_HU. This base class is already implementing the methods of the interface /VPCOE/IF_UPH_ENTITY_PROC and only the mapping of the HU must be provided by implementing the method /VPCOE/IF_UPH_ENTITY_HU_PROC~MAP_HU_DATA.

Sample BAdI implementation is provided with entity processor class /VPCOE/CL_HU_EXMPL.

The transaction code /VPCOE/UPH_PCI_HU belongs to this report execution.

Upload Handling Units from PLM

Initial Load Test Mode

Source System: ECC_EMEA
RFC Destination: VPCOE_RDP_PLM

Lookback Period End Date	29.02.2024
Lookback Period Duration	3
Material Number	to
Material Type	to
Material Group	to
Plant	to
Sales Organization	to
Delivery Type	to
Ship-to Party	to
Ship-to Country	to
Item Category	to
Distribution Channel	to
Division	to
Delivery Number	to
Delivery Plant Country	to

Report Parameters

Please consider the following when using report selection parameters:

- ‘Initial Load’: using this option all the HU of specified product(s) and for specific time period are extracted
- ‘Test mode’: Report use in test mode means that updating packaging composition with additional items will be omitted.
- ‘Source System’: the source system ID created on RDP configuration for packaging composition item entity.
- RFC Destination: The parameter should have the customer implemented target destination as default. If the customer does not have custom implementation for target RFC destination, then VPCOE_RDP_PLM is added as default value but this can be adjusted by the user. This gives customers more flexibility in choosing a destination name and avoids potential name conflicts.
- ‘Lookback Period End Date’/‘lookback period duration’: combination of these 2 fields represent selection Period for which deliveries with Handling Units are considered. Start date is pre-defaulted with current date. Lookback Period Duration represent number of months to restrict the selection back. This field is restricted up to 6 months maximum. Using this combination of fields, will give customers flexibility to do testing and understand what is the best number of months to look back.
- ‘Material number’: User can restrict selection for specific product(s) for which deliveries will be considered and finding it’s additional packaging data.

Because of the varieties of customizing on HU, the report is offering a sample Badi implementation covering:

- Extracting data for additional packaging composition items from handling units.
In this step, the report finds all the deliveries issued within the specified Lookback Period, which were packed using Handling Units. Please, note that in standard logic only HU items with item category HUPM and LEIH are considered. Please, adjust the logic adding your custom item categories if any. Then, packaging materials are allocated to their respective products and average use of packaging per product is calculated.
- Identifying the packaging composition from RDP that need to be enriched.
In this step, all existing packaging compositions for the products which were found in deliveries from the above step will be identified. If no packaging composition for a product is found in RDP, however, additional packaging composition items are calculated from HU data, no new packaging composition will be created in RDP, but the corresponding message will be added to the log output of the report.
- Replicating additional items to RDP for specific Packaging Composition.
In this step, Packaging Compositions are updated on RDP with additional packaging composition items and calculated quantities.

Class Overview

Report	/VPCOE/R_UPH_HU_DATA_LOAD	Extract Packaging Composition Items from Handling Units
Factory Class	SAME - Add new transfer mapper	
BADI	Call right upload entity from the entity processor	GC_UE_PCI_HU
WRAP_SPEC	No Change Needed	
WRAP_VA	No Change Needed	
Implementing Class	/VPCOE/CL_EXMPL_UPH_CUSTOM	Call the right example implementation;
Example Implementation	/VPCOE/CL_HU_EXMPL	Custom/Example Implementation for Extraction of Packaging Element from HU
Logger class	No change	

Extensibility in packaging data reports

The upload of packaging data from PLM (Product Lifecycle Management) requires that the according data exists in the Environment, Health, and Safety specification database. In addition, an Upload Entity Processor implementation for the respective Upload Entity PE_PLM and PC_PLM exist in the system and is returned by the BAdI implementation of /VPCOE/BADI_UPH_CUSTOM.

The upload of packaging composition data from recipes requires that the accompanying data exists in Recipe Management. In addition, an Upload Entity Processor implementation for the Upload Entity PC_RCP exists in the system and is returned by the BAdl implementation of /VPCOE/BADI_UPH_CUSTOM.

The upload of packaging composition data from BoM requires that the according data exist in the bill of material. In addition, an Upload Entity Processor implementation for the respective Upload Entity PC_BOM exists in the system and is returned by the BAdl implementation of /VPCOE/BADI_UPH_CUSTOM.

The upload of packaging elements data from Material Classifications requires that the according data exist in the material classification. In addition, an Upload Entity Processor implementation for the respective Upload Entity PC_MCL exists in the system and is returned by the BAdl implementation of /VPCOE/BADI_UPH_CUSTOM.

The upload of packaging composition data from Handling Units requires that the according data exist in handling units. In addition, an Upload Entity Processor implementation for the respective Upload Entity PC_HU exists in the system and is returned by the BAdl implementation of /VPCOE/BADI_UPH_CUSTOM.

Which entity processors are used for the corresponding Upload Entity is controlled by the BAdl /VPCOE/BADI_UPH_CUSTOM (Enhancement Spot /VPCOE/ES_UPH_CUSTOM). The BAdl mechanism allows to exchange a processor if needed, e.g. if customer specific data structures must be read.

Further, the BAdl implementation controls the used destination configuration. Destinations are possible for:

1. The source of the Packaging data (RFC destination to another system)
2. The target endpoint of the RDP Push API

VPCOE/BADI_UPH_CUSTOM – technical documentation

The Business Add-In (BAdl) /VPCOE/BADI_UPH_CUSTOM that is assigned to Enhancement Spot /VPCOE/BADI_UPH_CUSTOM is used during the upload of packaging related data to the API of SAP Responsible Design and Production (RDP). Its main task is to provide an extensibility mechanism to integrate customer specific implementations for extracting and processing of packaging related data.

This packaging related data is in the following called **Upload Entity**. An Upload Entity consists of:

- An RDP business object, for example, a packaging element with all its defined child nodes, such as, packaging fractions, product assignments and so on.
- The area of extraction, for example, Plant Maintenance (PLM) or Recipe Development (RD).

The following Upload Entities are currently defined in the system:

- **PE_PLM**: Packaging elements from PLM
- **PC_PLM**: Packaging compositions from PLM
- **PC_RCP**: Packaging compositions from Recipe Development
- **PC_BOM**: Packaging compositions from Bill of Material (BOM)
- **PE_MCL**: Packaging elements from Material Classification
- **PE_HU**: Packaging Compositions Items from Handling Units

For each Upload Entity there is a corresponding report to start the data upload.

When one of these reports is executed, the mentioned BAdI is called. As a result of the call, an instance of a customer specific implementation for the extraction and processing of the Upload Entity data must be returned. This instance is called Upload Entity Processor.

Handling Multi-Level BOMs in Packaging Composition Upload

Supported Depth for Subordinate Packaging Elements

- **SAP Responsible Design and Production (RDP)** currently supports only **one level** of subordinate packaging elements.
- As a result, the **example entity processor mapping implementation** used in the **Packaging Composition Upload Report for BOM** processes **only the first level** of the assembly BOM.
- Deeper levels beyond **level 1** are **not processed** by the standard implementation.

Handling Multi-Level BOMs (Level 1+N)

- If a **multi-level BOM** contains relevant subordinate packaging elements beyond the first level (**level 1+N**), this scenario is **not handled** by the standard logic.
- To accommodate such cases, a **custom mapping implementation** should be developed to:
 - Traverse deeper levels of the BOM hierarchy.
 - Extract and map relevant subordinate packaging elements beyond the first level.
 - Ensure accurate transfer of all necessary data to RDP.

Custom Mapping Considerations

- When implementing a **custom mapping logic**:
 - Ensure that all relevant subordinate elements beyond the first level are captured.
 - Adapt the mapping logic to correctly represent the additional levels of subordinate packaging elements in the JSON payload.
 - Test the implementation thoroughly to validate the correct transfer of data to RDP.

Prerequisites for Uploading Packaging Element and Packaging Composition Data

To successfully upload packaging element or packaging composition data from different sources to SAP Responsible Design and Production (RDP), the following prerequisites must be met:

1. Data Availability in Source Systems

Ensure that the required data exists in the respective source systems before initiating the upload:

- PLM (Product Lifecycle Management):
 - For uploading packaging composition data from PLM, the corresponding data must exist in the Environment, Health, and Safety (EHS) specification database.
- Recipe Development:
 - For uploading packaging composition data from Recipe Development, the corresponding data must be available in the Recipe Development system.
- Bill of Materials (BOM):

- For uploading packaging composition data from a BOM, the relevant data must exist in the Bill of Materials maintained in the system.
- Material Classification:
 - For uploading packaging element data from Material Classification, the required data must be classified in the material master data.

2. Upload Entity Processor Implementation

- An Upload Entity Processor implementation must exist in the system for the respective Upload Entity.
- This implementation ensures that data is correctly processed and transferred to RDP.

3. BAdI Implementation for Upload Entity Processor

- The BAdI responsible for providing the Upload Entity Processor instance must be implemented.
- A sample BAdI implementation is provided for reference, which demonstrates how to configure and extend the processor logic.

4. RFC Destination and OAuth Configuration

- The RFC destination and OAuth configuration must be properly maintained in the system to enable communication with the RDP Push API.

5. BAdI Standard Settings and Configuration

- For more information on standard settings related to this BAdI (such as filters, single or multiple usage scenarios), refer to the Spot Element Definitions tab in the BAdI Builder using transaction SE18.

BAdI Methods

The BAdI definition is active in the Integration accelerator package. It contains the following methods from interface `/VPCOE/IF_UPH_CUSTOM`:

`· get_entity_processor:`

This method returns which entity processor is used for the corresponding upload entity. The BAdI mechanism allows to exchange a processor if needed, for example, if customer specific data structures must be read.

In general, the implementation of this method will return a customer specific Upload Entity Processor implementation (that implements the `IF_SURDP_UPH_ENTITY_PROC` interface) for each Upload Entity type.

`· get_target_endpoint_dest:`

This method will return the end point destination of the target system. If this method is defined, the returned value will be displayed as the default RFC destination in the report selection screens for all the upload entities. If no specific value is returned, the default destination `RDP REPL API` is used.

BAdI Implementations

The BAdI implementation must be defined by the user and is not provided by SAP. As example see class `/VPCOE/CL_EXMPL_UPH_CUSTOM`.

The custom Upload Entity Processors must implement the `/vpcoe/if_uph_entity_proc` interface. To ease the definition of custom entity processors, there are base classes available which provide already necessary functionality:

- `/VPCOE/CL_UPH_PROC_BASE_PLM` for the extraction out of PLM
- `/VPCOE/CL_UPH_PROC_BASE_RCP` for the extraction out of Recipe Development
- `/VPCOE/CL_UPH_PROC_BASE_BOM` for the extraction out of Bill of Material
- `/VPCOE/CL_UPH_PROC_BASE_MCL` for the extraction out of Material Classification
- `/VPCOE/CL_UPH_PROC_BASE_HU` for the extraction from Handling Units

Therefore, custom Upload Entity Processors should inherit from these base classes. For examples of the custom Upload Entity Processors please have a look at the following classes:

- `/vpcoe/lcl_exmpl_cust_cmp_plm` for packaging composition extraction out of PLM
- `/vpcoe/lcl_exmpl_cust_cmp_rcp` for packaging composition extraction out of Recipe Development
- `/vpcoe/lcl_exmpl_cust_cmp_bom` for packaging composition extraction out of Bill of Material
- `/vpcoe/lcl_exmpl_cust_pe_plm` for packaging element extraction out of PLM
- `/vpcoe/lcl_exmpl_cust_pe_mcl` for packaging element extraction out of Material Classification
- `/vpcoe/cl_hu_exmpl` for packaging composition items extraction from handling units

Common class overview

Interface/Class	Description
<code>/VPCOE/CL_UPH_EXEC_LOAD_WRAP</code>	Wrapper Class for the Execution of an Entity Upload (used by the report)
<code>/VPCOE/IF_UPH_ENTITY_PROC</code>	Interface for Upload Entity Processors
<code>/VPCOE/CL_UPH_FACTORY</code>	Static Factory class, which provides the Upload Entity Processor implementations per Upload Entity
<code>/VPCOE/IF_UPH_CUSTOM</code>	BADI Interface for custom aspects of the data upload
<code>/VPCOE/BADI_UPH_CUSTOM</code>	BADI Definition for custom aspects of the data upload.
<code>/VPCOE/IF_UPH_LOGGER</code>	General Logger Interface
<code>/VPCOE/CL_UPH_LOGGER</code>	Default Logger implementation
<code>/VPCOE/CL_UPH_PROC_BASE</code>	Base (abstract) Upload Entity Processor class. Concrete Entity Processors implementations should inherit from this base class.

Messages

The Upload report shall output the following messages (if present: when the condition defined in column rule is met).

Message class: `/VPCOE/PLM`

API Version Selection for Packaging Data Integration

The RDP Integration Accelerator supports both V1 and V2 versions of the Packaging Data API. The version to be used can be explicitly defined in the entity processor class within the BAdl implementation.

Key Considerations:

- For V1 API: The processor class must be specified directly.
- For V2 API: The mapper handler class should be specified using the `GET_ENTITY_MAPPER` method.

Configuration Details:

API Version	Example Class for Implementation
V1 API	/VPCOE/CL_UPH_TM_PCKG_CMP_V1 (Packaging Composition)
	/VPCOE/CL_UPH_TM_PCKG_ELEM_V1 (Packaging Elements)
	/VPCOE/CL_UPH_TM_HU (Handling Units)
V2 API	/VPCOE/CL_TM_PCKG_CMP_V2 (Packaging Composition)
	/VPCOE/CL_UPH_TM_PCKG_ELEM_V2 (Packaging Elements)

Customization and Adjustments:

- A sample processor class is provided by default, but it can be replaced if further adjustments are required.
- If customization is needed, the processor class can be substituted with a custom implementation tailored to specific business needs.
- In the `GET_ENTITY_MAPPER` method, users can configure the appropriate class based on the API version selected.

This flexibility ensures seamless integration with both API versions while allowing for business-specific adjustments as needed.

Packaging fee replication

SAP Responsible Design and Production (RDP) is calculating packaging fees, which needs to be considered in the pricing procedures and invoices in the SAP ERP system. Therefore, RDP provide the possibility to transfer packaging fees as pricing condition records and material classification data in ERP system.

To be able to create these invoices, data needs to be replicated regularly from SAP Responsible Design and Production to your source system. The data extraction for this purpose is possible through Packaging Fees Retrieval report. Packaging Fee Posting report allows you to fill the according data into Pricing Conditions and Material Classification.

The RDP Packaging Fees Integration:

The RDP Packaging Fees Integration facilitates the retrieval and processing of packaging fee data from SAP Responsible Design and Production (RDP) into SAP ECC/S4H. It includes multiple reports and a BAdI framework for mapping and processing data according to business needs.

1. Reports for Packaging Fee Processing

Report Name	Description	Tcode
/VPCOE/R_PCKF_RETRIEVAL	Retrieves packaging fees from RDP.	/VPCOE/PCKF_RETRIEVE
/VPCOE/R_PCKF_POSTING	Converts transferred packaging fees into pricing condition records and/or material classifications.	/VPCOE/PCKF_POSTING
/VPCOE/R_PCKF_CUSTOM_EXEMPT	Retrieves customer exemptions.	
/VPCOE/R_PCKF_COMP_CODE_EXEM	Retrieves company code exemptions.	
/VPCOE/R_PCKF_CUSTOMER_ROLE	Retrieves customer roles.	

2. BAdI Implementation for Packaging Fee Processing

- A sample BAdI implementation is provided for converting transferred packaging fees into pricing condition records and material classifications.
- The sample implementation covers:
 - Customer-dependent and customer-independent SD (Sales and Distribution) and MM (Materials Management) pricing conditions.
 - Flexible mapping of data for pricing and classification structures.

3. Customization and Mapping

- When converting transferred packaging fees into pricing conditions and material classification data, a BAdI implementation is required for mapping.
- Due to the high degree of flexibility in pricing configurations and material classifications, SAP does not provide a default implementation.
- Users can customize the mapping logic as per their business-specific pricing and classification requirements.

Prerequisites to use the reports:

- The BAdI is implemented and provides the Entity Processor instance (custom class or base class) for the Entity Types PCKG_FEE and APL_BUSP.
- The RFC destination and the OAuth configuration are maintained in the system to access the RDP API.
- Definition of condition tables, condition types, access sequence, pricing procedures

When one of these reports is executed, the BAdI implementation is called. As a result of the call, an instance of an (customer specific) Entity Processor implementation must be returned.

Example implementation class: /VPCOE/CL_EXMPL_PCKF_CUSTOM

The example implementation class of the BAdI contains an entity processor implementation (commented out) for the Packaging Fee entity provided by a local class (lcl_exmpl_pckg_fee).

For logging in custom entity processor implementation, use application Logging Object /VPCOE/RDP, Sub-object PCKG_FEE.

Packaging fee retrieval

The Packaging Fee retrieval report /VPCOE/R_PCKF_RETRIEVAL is handling the retrieval of the entity data provided by the RDP Packaging Fee API.

Retrieval of Packaging Fees

Initial Load

Delta Load

Test Mode

RFC-Destination	VPCOE_RDP_PCKF
Package size	500

Use Filter Criteria

Report Configuration	
Report Category	
Country	
Context	RDP_DEFAULT
Business Process Direction	

Extract fees for customer independent/dependent scenario

Dependent Scenario
 Independent Scenario

To use the support for customer role dependent fees in the API, it is necessary to select dependent scenario which will set the includeCustomerRoleDependentFees parameter in request to 'true'.

If the parameter includeCustomerRoleDependentFees is set to 'false' means the attribute ShipToPartyRole is empty and only customer role independent fees are returned. If the parameter is set to 'true', the attribute ShipToPartyRole is populated with the relevant customer role or is left empty if there is no role specified. For more information about available customer roles, see [Customer Role](#).

Example implementation for creating the pricing conditions for customer dependent and independent scenarios are included in sample badi implementation class /VPCOE/CL_EXMPL_PCKF_CUSTOM.

For the API documentation, see [API Reference](#).

For details on the packaging fee extracted data please have a look here: [Extracted Data | SAP Help Portal](#).

Packaging fee posting

The Packaging Fee Posting report /VPCOE/R_PCKF_POSTING is responsible to use the entities (calculated tax data) from the RDP for creating the pricing conditions and convert the product data to Material Classification for the product the individual record belongs to.

Packaging Fees available for posting:	186
<input type="checkbox"/> Test Mode	
Additional Settings <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;"> Retain failed records in days 7 </div>	

When executing Packaging Fee Posting report:

- The *Posting* report attempts to post all retrieved packaging fees (previously saved in cache table)
- If any errors occurred during the posting, failed records remain in the cache for the number of days specified in the *Retain failed records in days* field. This allows users to review and analyze errors.
- If this field is set to *0 days*, the failed records are deleted immediately from the cache/temporary table.

For packaging fee posting, customer is responsible to:

- Definition of condition tables, condition types, access sequence, pricing procedures
 - Define concept for SD & MM Conditions for Packaging Fees
 - Create Condition Table(s)
 - Add Condition Table(s) to Access Sequence
 - Assign Access Sequence to Pricing Condition Type
 - Add Condition Type to Pricing Procedure
- Implement available BAdI (Copy Sample BAdI provided) and adjust the filling for packaging fees into condition records
 - Define filters based on which the data shall be replicated
 - Extend the mapping with the exact pricing condition tables
 - Logic to split the data to multiple ERP systems (optional)

Replication of Customer Exemptions

The Customer exemptions retrieval report `/VPCOE/R_PCKF_CUSTOM_EXEMPT` is handling replication of customer exemptions which can be used for the creation of customer and report fraction-based sales and distribution pricing conditions for exemptions. Those replications are currently applicable only for Extended Producer Responsibility (EPR) reporting solution for Italy (CONAI). For more information about maintaining customer and company code exemptions, see [Report Category Extension](#).

Note: If the Customer is blocked, it will not be returned in the response, even if it is applicable to the respective PackagingComposition. If the blocked Customer is the only one applicable to the PackagingComposition, then the fee with this packaging composition will be skipped from the replication.

Replication of CompanyCode Exemptions

The Company Code exemptions retrieval report `/VPCOE/R_PCKF_COMP_CODE_EXEM` is handling replication of company code exemptions which can be used for the creation of company code-based material management pricing conditions for exemptions. Those replications are currently applicable only for Extended Producer Responsibility (EPR) reporting solution for Italy (CONAI). For

more information about maintaining customer and company code exemptions, see [Report Category Extension](#).

Replication of Customer Roles

The Customer Roles retrieval report `/VPCOE/R_PCKF_CUSTOMER_ROLE` allows the extraction of customer roles. Replication of customer roles to the SAP ERP systems is used for the identification and creation of the correct customer role and product-based sales and distribution pricing condition.

Extensibility in Packaging fee reports

The packaging fee retrieval and posting reports and also the exemptions replications and customer roles dependent fees replications are using the BAdI `/VPCOE/BADI_PCKF_CUSTOM`. The BAdI enables the customer to implement customer specific extensions.

Interface/Class/Report	Description
<code>/VPCOE/ES_PCKF_CUSTOM</code>	Enhancement Spot for BAdI
<code>/VPCOE/BADI_PCKF_CUSTOM</code>	BAdI Definition
<code>/VPCOE/IF_PCKF_CUSTOM</code>	BAdI Interface
<code>/VPCOE/CL_EXMPL_PCKF_CUSTOM</code>	Example Implementation for BAdI Interface

Interface `/VPCOE/IF_PCKF_CUSTOM`:

Method	Description
<code>GET_ENTITY_PROCESSOR</code>	Returns the entity processor instance for the given entity type.
<code>GET_TARGET_ENDPOINT_DEST</code>	Gets the default destination name. This method can return the destination name which defines a http target to the RDP Test Client.

For logging in custom entity processor implementation, use application Logging Object `/VPCOE/RDP`, Sub-object `PCKG_FEE`.

`/VPCOE/BADI_PCKF_CUSTOM` – technical documentation

The Business Add-In (BAdI) `/VPCOE/BADI_PCKF_CUSTOM` that is assigned to enhancement spot `/VPCOE/ES_PCKF_CUSTOM` is used during the download and processing of packaging fee data from the API of SAP Responsible Design and Production (RDP). Its main task is to provide an extensibility mechanism to integrate customer specific implementations for the retrieval and posting of packaging fee data.

Before describing this extension mechanism in more detail, it is necessary to understand the concept of Entity Types and their respective Entity Processor implementations.

Entity Type and Entity Processor

The Packaging Fee API of SAP Responsible Design and Production (RDP) is providing different entities:

- Packaging Fee (`PCKG_FEE`)
- Applicable Business Partners (`APL_BUSP`)

These entities are named in the following as Entity Type. The processing of the corresponding entity data of an Entity Type is implemented by a so-called Entity Processor class. An Entity Processor class must implement the interface `/VPCOE/IF_PCKF_ENTITY_PROC` that contains the following important methods:

- INIT_PROCESSOR: This method is used to initialize the Entity Processor
- PREPARE_RETRIEVE: The preparation steps for the entity data retrieval are implemented in this method.
- RETRIEVE_PACKAGE: The implementation of the entity data retrieval is implemented in this method. For example, the retrieval of the entity data from the corresponding RDP API must be implemented in this method.
- TRANSFER_PACKAGE: The transfer of the entity data to its target is implemented in this method. For example, the conversion of Packaging Fee entity data to ERP pricing conditions must be implemented here in the Entity Processor class for Entity Type PCKG_FEE.

For each Entity Type there is already a base class available in the system that must be derived for a custom logic:

- /VPCOE/CL_PCKF_PROC_PCKG_FEE: base class for Entity Type PCKG_FEE
- /VPCOE/CL_PCKF_PROC_APPL_PART: base class for Entity Type APL_BUSP

The data retrieval from the RDP API (method PREPARE_RETRIEVE and RETRIEVE_PACKAGE) is already implemented in these base classes.

But for the implementation of the customer specific conversion logic of Packaging Fee entity data to ERP pricing conditions, a custom implementation must be provided by deriving from class /VPCOE/CL_PCKF_PROC_PCKG_FEE and overriding the method TRANSFER.

BAdI Methods

The BAdI definition is active in the standard system. It contains the following methods from interface /VPCOE/IF_PCKF_CUSTOM:

- get_entity_processor:
This method returns which entity processor is used for the corresponding entity type. The BAdI mechanism allows to exchange a processor, for example, when a customer specific conversion of Packaging Fee data to ERP Pricing Conditions is needed.
If no custom implementation is returned, the Entity Processor base class is used instead.
- get_target_endpoint_dest:
This method will return the end point destination of the RDP source system. If this method is defined, the returned value will be displayed as the default RFC source in the retrieval report selection screen. If no specific value is returned, the default destination VPCOE_RDP_PCKF is used.

BAdI Implementations

The BAdI implementation must be defined by the user and is not provided by SAP. As an example, class /VPCOE/CL_EXMPL_PCKF_CUSTOM is linked as a BAdI example implementation.

For more information about settings for a BAdI, such as filters, single or multiple uses, see the Spot Element Definitions tab in the BAdI Builder (transaction SE18).

Example implementation class: /VPCOE/CL_EXMPL_PCKF_CUSTOM

The example implementation class of the BAdI contains an entity processor implementation (commented out) for the Packaging Fee entity provided by a local class (lcl_exmpl_pckg_fee).

During the posting, the packaging fee records are converted to pricing conditions of Condition Type ZRDP. Since this condition type currently holds 3 different condition tables (identified by separate keys) the according condition table (key) is selected with regards to the available organizational data in the packaging fee.

HOW TO CHECK REPLICATION RESULTS

When running the RDP reports, the results are displayed on the screen and also are logged in SLG Object '**/VPCOE/RDP**'.

Results on the screen

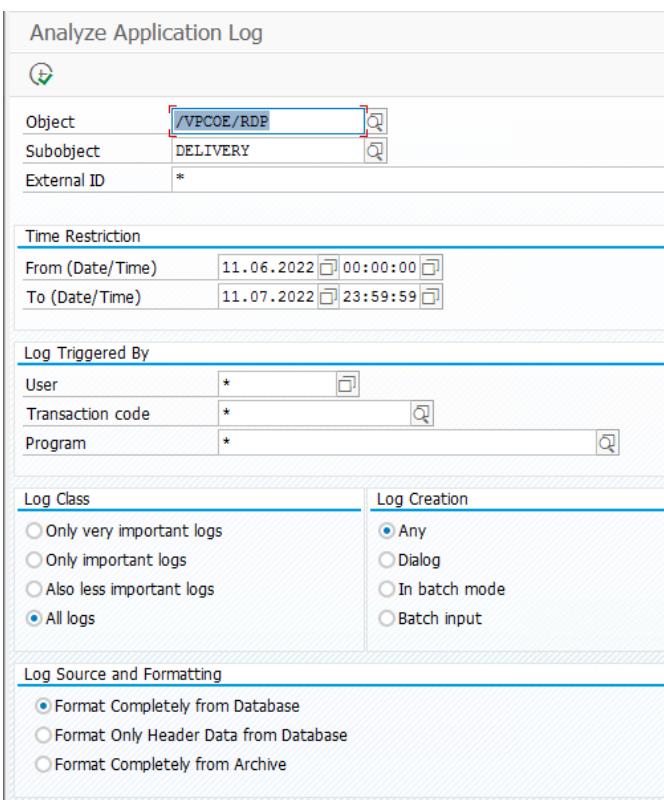
When running any RDP integration report in foreground mode the replication results are visible on the screen in a similar way as below:



```
Send Delivery Data to RDP
Send Delivery Data to RDP
DELIVERY -
Start to Process Package - 1
15 entries sent successfully
A total of 15 records were sent.
```

Application log results

- After running the RDP Integration reports you can see the results on the logged object (even when running in background mode):
 1. Open transaction 'SLG1'. Use object '**/VPCOE/RDP**'.
 2. For 'Subobject' use the required service ID or '*'. For packaging fee replication logs, use the sub-object '**PCKG_FEE**' .



Analyze Application Log

Object: /VPCOE/RDP

Subobject: DELIVERY

External ID: *

Time Restriction

From (Date/Time): 11.06.2022 00:00:00

To (Date/Time): 11.07.2022 23:59:59

Log Triggered By

User: *

Transaction code: *

Program: *

Log Class

Only very important logs

Only important logs

Also less important logs

All logs (selected)

Log Creation

Any (selected)

Dialog

In batch mode

Batch input

Log Source and Formatting

Format Completely from Database (selected)

Format Only Header Data from Database

Format Completely from Archive

Display logs								
Date/Time/User	Nu...	External ID	Object text	Sub-object text	Transac...	Program	Mode	Log number
> 16.06.2022 14:45:28 C5318949	3		VP CoE RDP Log	DELIVERY	SEU_INT /VPCOE/D...	Dialog pro...	00000000000003528224	
> 28.06.2022 16:15:26 C5298984	3		VP CoE RDP Log	DELIVERY	/VPCOE/D...	Dialog pro...	00000000000003659491	
.....								
 Ty... Message Text  Start to Process Package - 1  Number of entries send successfully - 6 entries  Total number of records sent successfully - 6 records.								

KNOWN TECHNICAL ISSUES AND TROUBLESHOOTING

Technical Issue/ restriction	Solution	Valid from / to SAP Software Components
Errors when installing the general object CR, due to missed objects in standard code (ex. MEINS_D being not available in the system)	SAP Note 2526405 - /UI2/CL_JSON Corrections released in 2018 should be implemented	SAP_UI 750/751/752 UI_700 200
Multilanguage capabilities are not supported (only English language is supported for now)		

www.sap.com/contactsap

© 2025 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See www.sap.com/trademark for additional trademark information and notices.

